Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Rok Petek

Optična razpoznava znakov v slikah naravnih scen

magistrsko delo

študijski program druge stopnje
računalništvo in informatika

Mentor: doc. dr. Luka Šajn

Ljubljana, 2016

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Rok Petek, z vpisno številko **63090366**, sem avtor diplomskega dela z naslovom:

Optična razpoznava znakov v slikah naravnih scen

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Luka Šajna,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 22. marec 2016 Podpis avtorja:

Posebej bi se rad zahvalil svojim staršem in vsem ostalim, ki so mi tekom študija stali ob strani in me spodbujali pri delu.

Zahvalil bi se tudi mentorju doc. dr. Luki Šajnu, za dobre nasvete in motivacijo pri delu.

Kazalo

Seznam uporabljenih kratic in simbolov

ANN - Artificial Nural Network; umetna nevronska mreža
CO-HOG - Co-occurence of Histogram of Oriented Gradients; deskriptor značilk sopojavitvene matrike
CVL OCR DB - Computer Vision Laboratory OCR DataBase; OCR podatkovna baza Laboratorija za računalniški vid
HOG - Histogram of oriented gradients; deskriptor značilk
ICDAR - International Conference on Document Analysis and Recognition; mednarodna konferenca razpoznave in analize dokumentov
K-NN - K-Nearest Neighbours; k-najbližjih sosedov
NB - Naive Bayes; naivni bajes
OCR - Optical Character Recognition; optična razpoznava znakov
PHOG - Pyramid of Histogram of Oriented Gradients; deskriptor značilk piramidnega sistema
SVM - Support Vector Machine; metoda podpornih vektorjev

Povzetek

V magistrskem delu so predstavljene in opisane sodobne metode optične razpoznave znakov v slikah naravnih scen. Izbrane so bile metode, ki dosegajo visoko točnost in so robustne na osvetlitev ter ostale geometrijske spremembe. Naše delo temelji na implementaciji treh različnih metod za pridobivanje značilk. Osnovna metoda HOG, na kateri temeljita tudi ostali dve metodi je ena izmed bolj popularnih metod pridobivanja značilk. Metoda HOG je bila primarno uporabljena pri detekciji ljudi, vendar je pri razpoznavi znakov v slikah naravnih scen modificirana, za doseganje boljših rezultatov. Na metodi HOG bazira metoda PHOG, ki pretvori osnovni HOG v piramidalni sistem ter obenem vključuje bilinearno interpolacijo. Zaradi piramidne strukture PHOG, je ta metoda počasnejša od metode HOG, vendar bolj natančna, saj je tudi vektor značilk večji. Tretja metoda, ki smo jo implementirali, je metoda Co-HOG, ki od metode HOG podeduje vse dobre lastnosti, kot je invariantnost na različno svetlost in lokalne geometrijske spremembe. Co-HOG se razlikuje po tem, da značilke vsebujejo tudi prostorska razmerja med slikovnimi elementi, s čimer se znak bolj natančno opiše in razpozna, med drugim je tudi hitrejša metoda pridobivanja značilk.

Zaradi različnih naravnih faktorjev v slikah znakov naravnih scen je razpoznava znakov s tradicionalnimi sistemi optične razpoznave znakov nenatančna, saj ti sistemi predpostavljajo, da se znaki ne razlikujejo v pisavi, barvi in ozadju znaka. Pri pridobivanju robustnih značilk znakov naravnih scen se uporablja metode, ki so invariantne na velikost znaka, šum ozadja, tip pisave ter na vizu-

alne efekte, ki pritegnejo pozornost, kot je npr. prelivanje barv v posameznem znaku. Zgoraj opisane metode ne potrebujejo klasičnega predprocesiranja in binarizacije slike znaka, kot to počnejo tradicionalni sistemi, saj te zajemajo značilke z metodami, ki opišejo videz objekta in obliko z intenziteto gradientov ter smermi robov.

Metode pridobivanja značilk so bile evalvirane na različnih podatkovnih bazah, kot so ICDAR, Chars74K, CVL OCR DB. Generirali smo tudi sintetično podatkovno bazo, ki imitira znake v naravnih scenah, tako da vključuje množico različnih pisav ter šumov v slikah. Sintetična podatkovna baza znakov je bila generirana z namenom povečanja učne množice ter izboljšanja rezultata klasifikacijske točnosti.

Ključne besede:

Optična razpoznava znakov, naravne scene, računalniški vid, HOG, PHOG, Co-HOG, SVM, ANN, K-NN

Abstract

This masters thesis presents and describes modern methods of optical character recognition in natural scenes. Methods with high classification results and are robust to illumination and geometric transformations were selected for the thesis. Our work is based on the implementation of three different methods for obtaining features. The basic HOG method, which also underlies the other two methods is one of the most popular feature extraction methods in object detection and character recognition. HOG method was primarily used in connection with human detection, but was adapted for character recognition also. PHOG method, which is based on HOG, converts the basic HOG algorithm into a pyramid scheme and also includes bilinear interpolation. Due to the pyramid structure of PHOG, the method is slower than the HOG algorithm, but more precise, since the feature vectors are larger. The third feature extraction method, which we have implemented is Co-HOG algorithm, which inherits all the good qualities of HOG method, such as invariance to illumination and geometric changes. Co-HOG is differs from HOG and PHOG, by its feature representation, where it also captures the spatial relationship of neighbouring pixels in order to describe the character more accurately. Among other things Co-HOG is also a computationally faster than HOG and PHOG.

Due to various factors in natural scene text images, the traditional character recognition systems produces inaccurate results, because it assumes that the characters do not differ in fonts and colors and presumes a monotonous background of images, whereas in obtaining features from natural scene im-

ages, the algorithms should be robust and invariant to character sizes, background noise, different fonts, local illumination changes and visual effects that draw attention, such as color blending. The above described methods do not require preprocessing and segmentation as traditional systems do, since the extract features with methods that describe the appearance of the object and the shape with gradient intensity and edge directions.

Feature extraction methods were evaluated on a variety of databases such as ICDAR, Chars74K, CVL OCR DB. We have also generated a synthetic database of character images, that simulates characters in natural scenes, by including large variety of different fonts and noises in images. Synthetic image database was generated with the aim of increasing the training set and the improvement of classification accuracy.

Keywords:

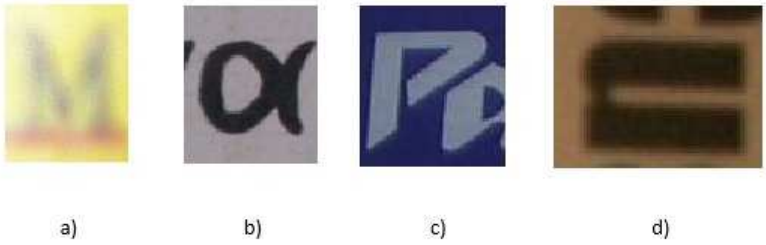
Optical character recognition, natural scenes, computer vision, HOG, PHOG, Co-HOG, SVM, ANN, K-NN

Poglavje 1

Uvod

Razpoznava znakov v slikah naravnih scen je specifično področje, kjer tradicionalne metode razpoznave znakov ne prinesejo željenih rezultatov. Med tradicionalne metode razpoznave znakov spadajo sistemi, ki razpoznajo znake iz skeniranih dokumentov, obrazcev, položnic, potovalnih listin, bančnih izpiskov itd., torej v primerih kjer so vnaprej znana in predvidena določena pravila, kot je položaj znakov, tip pisave, barva pisave, dobra osvetljenost, razločnost in ostali parametri, ki pripomorejo k bolj natančni razpoznavi.

Dokumentni OCR sistemi, kot so Tesseract [1] in Abbyy Fine Reader [2], so namenjeni predvsem iskanju informacij v veliki količini dokumetnih zapisov. Iz večjih podatkovnih zbirk kot je Googlova anotirana podatkovna zbirka teksta iz ulic (angl. Street View Text Database) [3] bi želeli najti določene informacije in iz njih razbrati kontekst npr. lokacijo ali ime objekta, česar ni mogoče z dokumetnimi OCR sistemi. Metode razpoznave znakov v slikah naravnih scen rešujejo ta problem tako, da se osredotočajo na večjo robustnost pridobivanja značilk. Internetni brskalniki med drugim ponujajo tudi opcijo iskanja besed v slikah, kar pomeni, da s pomočjo sistemov razpoznave znakov pravilno anotirajo sliko. Sodobni OCR sistemi, ki so namenjeni razpoznavi znakov v naravnih scenah, omogočajo lažje pridobivanje podatkov iz slik objektov, npr. če bi želeli poiskati določen model letala, bi ta lahko bil

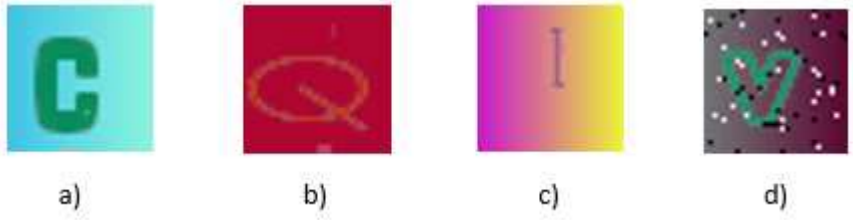


Slika 1.1: Primeri slabih znakov (a) zamegljena slika, (b) dvoumen znak, (c) več znakov na sliki, (d) rotiran znak.

detektiran s pomočjo primerjave slik s trenutnimi slikami modela letala, lahko pa bi preprosto razbrali ime modela iz slike.

Znaki v slikah naravnih scen so lahko različnih barv, vsebujejo šum v ozadju, tipi pisave so različni, osvetlitev ni enakomerna. Te nepravilnosti je torej potrebno odpraviti na robusten način in zajeti strukturo znaka ter najti podobnosti med različnimi oblikami znakov, s katerimi lahko razlikujemo znake. Metode razpoznave znakov v slikah naravnih scen delimo na dva segmenta, na tiste ki sliko predprocesirajo, ji odpravijo šum in sliko binarizirajo ter na tiste ki ne potrebujejo predprocesiranja. V magistrski nalogi smo se osredotočili na metode pridobivanja značilk, ki ne potrebujejo predprocesiranja in temeljijo na histogramu orientacij gradientov (HOG). Izbrali smo metode, ki ponujajo dovolj visoko hitrost razpoznave, torej da so primerne realnočasovnim aplikacijam ter obenem dovolj natančne.

Pridobljene značilke znakov smo testirali na različnih podatkovnih zbirkah slik znakov. Uporabili smo sledeče podatkovne zbirke: ICDAR (International Conference on Document Analysis and Recognition) [4], CVL OCR DB [5], CHARS74K [6] ter sintetično podatkovno zbirko, ki smo jo implementirali sami. Sintetična zbirka slik znakov je bila generirana z namenom razširitve učne množice, ki jo ponujajo že obstoječe podatkovne zbirke.



Slika 1.2: Primeri sintetičnih znakov (a) berljiv znak, (b) tanek tip pisave, (c) ozadje z gradientom, (d) šumno ozadje.

Pri implementaciji sintetične zbirke slik znakov smo želeli generirati znake, ki so čimbolj podobni znakom v slikah naravnih scen, ti so različnih tipov pisav, znaki so rotirani in nagnjeni, ozadja vsebujejo šume in gradiente. Kot je razvidno iz slike 1.2, smo se pri generiranju sintetičnih znakov osredotočali na poustvarjanje efektov in defektov v slikah naravnih scen.

Zadnji korak razpoznave znakov je klasifikacija znaka v pravilni razred. Klasifikacijo znakov smo razdelili na 52 razredov brez številk ter na 26 razredov, kjer smo v slednjem primeru upoštevali invariantnost velikosti znaka. Pri klasifikaciji smo uporabili več metod strojnega učenja in analizirali, katera prinese najboljše rezultate pri določenemi metodi pridobivanja značilk. Uporabili smo klasifikacijske metode kot so ANN, SVM, K-NN, NB ter za vsak klasifikacijski model definirali optimalne parametre, s katerimi smo lahko dosegali najvišjo klasifikacijsko točnost.

Poglavje 2

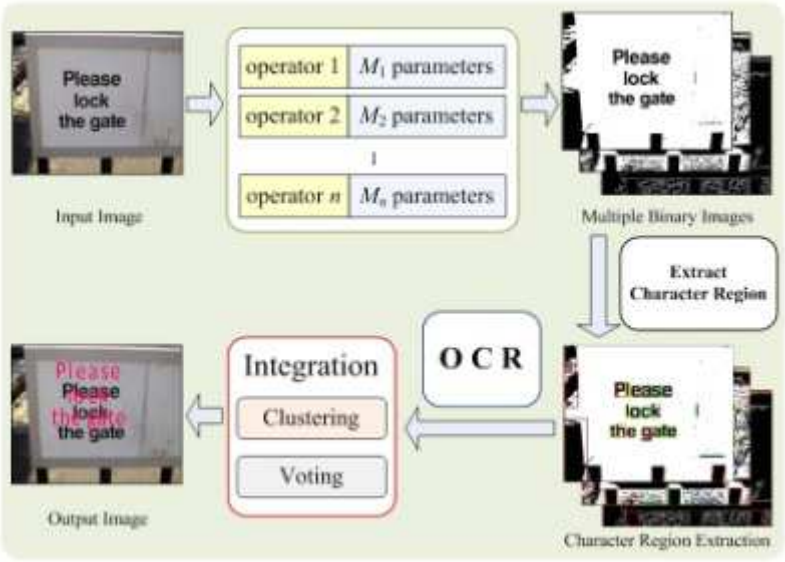
Pregled sorodnih del

Razširitev socialnih medijev in tehnologije posledično povzroča tudi razvoj optične razpoznave znakov. Uporabniki si poleg teksta izmenjujejo tudi slike, ki vsebujejo tekst. Ta trend vodi v neanotirano podatkovno bazo slik, katere je težko organizirati ali jih iskati v nekem kontekstu. Velik del teh slik vsebuje tekst, ki lahko predstavlja lokacijo prostora ali znamko avtomobila ali kaj drugega. Z OCR sistemi je take slike precej lažje anotirati, kot pa z algoritmi, ki delujejo na podlagi detekcije lokalnih značilnih točk v sliki (SIFT [7], katerega se je tudi uporabljalo za razpoznavo znakov). V slednjem primeru je potrebno najprej iz slik izvleči značilne točke in jih primerjati z že obstoječimi v podatkovni bazi. Veliko metod za razpoznavo objektov uporablja razpoznavo teksta v procesu, ki pripomore k razpoznavi končnega objekta [8].

Največji problem pri razpoznavi znakov v slikah naravnih scen ostaja nepredvidljivost okolja, v katero je postavljen znak, saj je ta lahko zamaknjen, zamegljen, nerazločnega tipa pisave, različnih barv, neenakomerno osvetljen, osenčen in drugo. Kvaliteten sistem razpoznave znakov v slikah naravnih scen mora torej biti robusten na te spremembe, ki se pojavljajo v nenadzorovanih okoljih.

Sitemi razpoznave znakov se delijo na dva dela in sicer na metode, ki potrebujejo predprocesiranje in meode, ki tega ne potrebujejo. V predprocesi-

ranje slike znaka spada odstranitev šuma, izostritev slike, segmentiranje oz. binarizacija, poprava naklona in perspektive. Vsak od teh problemov predprocesiranja je lahko sam po sebi zelo kompleksen in računsko zahteven, kar pri realnočasovni razpoznavi ni zaželjeno.

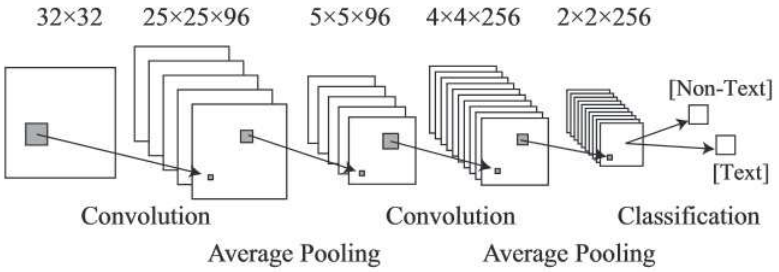


Slika 2.1: Diagram ogrodja večih hipotez. Povzeto po "Scene character detection and recognition with cooperative multiple-hypothesis" [9].

Metode brez predprocesiranja

Med metode, ki ne potrebujejo predprocesiranja spada tudi ogrodje večih hipotez 2.1 (angl. Multiple-Hypothesis Framework [9]). S pomočjo hevrističnih pravil, ki določajo omejitve na področjih segmentacije, razmerja slike, barvne neskladnosti, integracijski modul dopolnjuje trenuten OCR sistem ter odstrani nezaželjene detekcije in jih dopolni z manjkajočimi deli. Predlagano ogrodje dopolni vrzel med detekcijo in razpoznavo znakov v scenah, kjer OCR sistem učinkovito vpliva za izboljšanje rezultata. Poleg tega predlagana metoda dosega razpoznavo in detekcijo na ravni znaka, katera omogoča obravnavo na posameznem znaku, tekstu v poljubni orientaciji ali tekstu vzdolž krivulj [9].

Konvolucijske nevronske mreže (CNN [10]) spadajo med novejše metode globokega učenja (angl. deep learning), saj uporabljajo več deset skritih nivojev (angl. hidden layers). CNN metoda ne potrebuje predprocesiranja in celo določitve parametrov ne. CNN metode so bile predhodno uspešne tudi



Slika 2.2: Arhitektura CNN metode. Povzeto po "End-to-end text recognition with convolutional neural networks" [10].

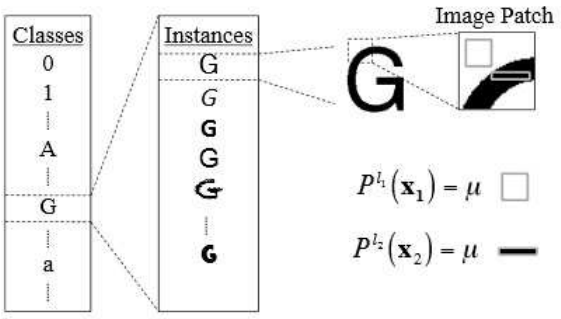
pri reševanju problema razpoznave lastnoročne pisave, razpoznave objektov ter optične razpoznave znakov v slikah naravnih scen.

Na sliki 2.2 lahko na prvem mestu vidimo vhodni podatek oz. sliko znaka, ki je normalizirana na velikost 32×32. Preostali segmenti prikazujejo konvolucijske ter podvzorčne nivoje in na koncu klasifikacijo. CNN metoda trenutno dosega najboljše rezultate na podatkovni zbirki ICDAR [4] in sicer 83.9 % klasifikacijsko točnost.

V podobnem delu [11] se CNN metoda uporablja tudi pri detekciji teksta v naravnih scenah na podatkovni zbirki slik ICDAR 2003 [4] v kombinaciji z lingvističnim znanjem, ki odstrani napačno detektirane znake. Za klasifikacijski model se v tem primeru uporablja CNN, ki prekaša metodo podpornih vektorjev SVM [12]

Prav tako se segmentaciji izogiba implementacija razpoznave ukrivljenega teksta v naravnih scenah. Ta metoda je implementirana na podlagi skritih markovskih modelov HMM [13] (angl. Hidden Markov Models). Metoda integrira drsno okno in ekstrahira značilke, ki so dostavljene HMM sistemu za razpoznavo.

Metoda Houghovih gozdov [14] se izogne procesu segmentacije z uporabo navzkrižnega skaliranja binarnih značilk (angl. cross-scale binary features), kot je razvidno na sliki 2.3.



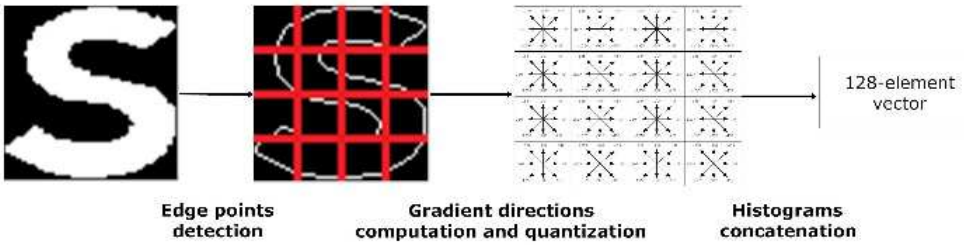
Slika 2.3: Navzrkižno skaliranje binarnih značilk. Povzeto po " Text recognition in natural images using multiclass hough forests" [14].

Metode s predprocesiranjem

Poleg omenjenih metod, ki ne potrebujejo predprocesiranja pa vseeno obstajajo metode, ki potrebujejo predprocesiranje ali segmentacijo. Ena izmed bolj naprednih metod, ki uporabljajo predprocesiranje in MSER značilke je ogrodje preverjanja hipotez [15], ki uporablja sintetično podatkovno zbirko slik za učenje algoritma, s čimer prihrani časovno potratno pridobivanje in označevanje realnih podatkov. Ta metoda izrablja maksimalno stabilne ekstremne regije (angl. Maximally Stable Extremal Regions - MSERs), ki zagotavljajo robustnost na geometrične in osvetlitvene pogoje. Razpoznavna točnost te metode na podatkovni zbirki slik ICDAR [4] je 74 %.

Metoda funkcij usmerjenih gradientov [16] (angl. Gradient Direction Features) spada v kategorijo metod razpoznave znakov, ki potrebujejo predprocesiranje slike. Ta metoda dobi za vhodni element že binarizirano sliko, na podlagi katere izračuna vektor značilk, katerega nato klasificira s pomočjo klasifikacijske metode k-najbližjih sosedov (K-NN). Metoda temelji na usmerjenih gradientih, kar je podobno, kar počne metoda HOG [17], saj najprej detektira robove binarizirane slike in nato izračuna smer gradienta za vsak robni piksel.

Slika je binarizirana, zato je gradient izračunan samo na robnih slikovnih enotah, kar pomeni hitrejši izračun. Smeri gradienta se nato umesti v

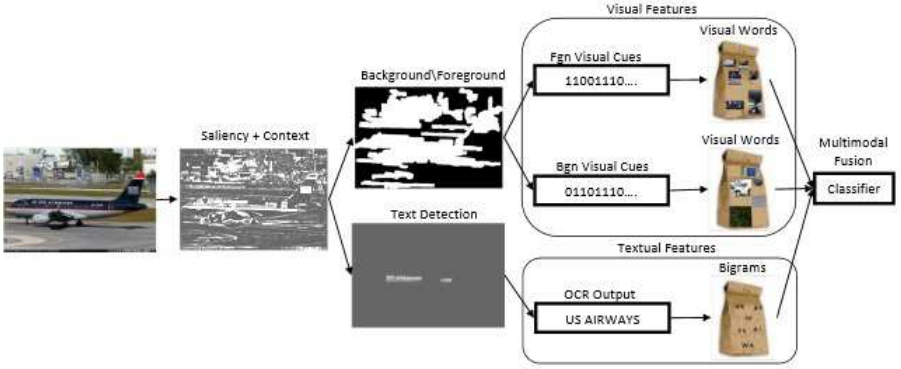


Slika 2.4: Funkcija usmerjenih gradientov. Povzeto po "A character recognition method in natural scene images" [16].

histogram z osmimi stolpci, oz. za posamezno orientacijo. Vhodna binarizirana slika je razdeljena na 16 blokov, s katerimi pridobi prostorske informacije znaka, nato je vsak blok konkateniran v 128 dimenzionalni vektor značilk 2.4. Ker slednja metoda temelji izključno na smeri robnih slikovnih elementov, nanjo ne vpliva niti barva niti intenziteta. Ta metoda dosega 68.2 % točnost na podatkovni zbirki slik ICDAR [4].

Naslednja metoda predlaga večkratno segmentacijo z uporabo naključnih Markovih polj [18] (angl. Markov random field - MRF). Večkratna segmentacija je tu opravljena na istih tekstovnih regijah, pri katerih se generira večkratna hipoteza binarnih tekstovnih slik. Segmentacijski algoritem je definiran kot statistično označevanje in temelji na Markovih naključnih poljih. Ozadje slike vsake hipoteze je odstranjeno z analizo povezanih komponent in vpeljuje strožje omejitve na sivinske vrednosti slike, s pomočjo robustnega 1-dimenzionalnega medianinega operatorja.

Kot smo omenili prej, se metode razpoznave teksta uporabljajo tudi pri razpoznavi objektov [19]. Primer aplikativnosti razpoznave znakov pri razpoznavi objektov je razvidna na sliki 2.5, kjer je tekst segmentiran in ekstrahiran iz slike objekta ter klasificiran glede na kontekst. Ta metoda uporablja za klasifikator SVM, v povezavi z vrečami besed (angl. Bag Of Words) in K-means gručenjem.



Slika 2.5: Detekcija objektov z OCR sistemom [19].

Med bolj pogoste metode razpoznave znakov v slikah naravnih scen spada metoda SIFT [20]. SIFT (angl. Scale-invariant feature transform), ki detektira lokalne značilke v sliki tako, da za vsak objekt na sliki detektira interesne točke, ki opisujejo celoten objekt. Opis objekta je lahko uporabljen pri identifikaciji enakega objekta na drugi sliki z drugačno vsebino. Najpomembnejša prednost algoritma SIFT je invariantnost na skaliranje, šum ter neenakomerno osvetljenost pri detektiranju podobnega objekta. Za efektivno razpoznavo znakov z metodo SIFT se uporablja gosti SIFT, ki ekstrahira značilke na enakomerno porazdeljeni mreži, s katero pridobimo uniformno število značilk.

Večina sodobnih in naprednih metod razpoznave znakov v slikah naravnih scen je neodvisna od predprocesiranja oz. segmentacije slike. Prav te metode dosegajo največjo klasifikacijsko točnost pri razpoznavi. V magistrski nalogi smo se na podlagi teh ugotovitev osredotočili na implementacijo metod, ki temeljijo na pridobivanju značilk brez predprocesiranja.

Poglavje 3

Uporabljena orodja

3.1 Python

Programski jezik Python [21] je večnamenski odprtokodni jezik, ki se ga med drugim uporablja za implementiranje spletnega zaledja, preprostih igric, strojno učenje, vedno bolj pogosta pa je njegova uporaba v bioinformatiki in računalniškem vidu. Python je objektno orientiran skriptni jezik vendar ga lahko uporabljamo v različnih načinih programiranja, kot je funkcijsko, imperativno ali proceduralno programiranje. Python skrbi sam za upravljanje s pomnilnikom, štetjem referenc in zbiranjem "smeti" (angl. garbage collection), obenem je tudi visoko razširljiv jezik in ponuja mnogo različnih razširitev za optimizacijo svojega delovanja (Cython [22], PyPy [23])

3.1.1 OpenCV

Za manipulacijo in branje slik smo uporabili knjižnico OpenCV [24], ki je bila izdana z BSD licenco in je brezplačna za akademske in komercialne namene. Knjižnica je napisana v optimiziranem C/C++ programskem jeziku, vendar ponuja vmesnike za programske jezike kot so C++, C, Java ter Python. Njegova arhitektura je zgrajena z močnim poudarkom na realno-časovnih aplika-

cijah. Knjižnjica ponuja visok nabor algoritmov in metod, ki so namenjene razpoznavi obrazov, razpoznavi gest, mobilni robotiki, identifikaciji objektov, detekciji objektov, navidezni resničnosti, segmentaciji itd., ponuja pa tudi statistično knjižnico, ki vsebuje algoritme za strojno učenje.

3.1.2 Matematični programski paketi

NumPy [25] je razširitveni paket za programski jezik Python, ki ponuja napredne matematične operacije nad velikimi, več-dimenzionalnimi matrikami in vektorji. Numpy omogoča hitro procesiranja večjih matrik in vektorjev, njegova zmogljivost je primerljiva z ekvivalentno kodo, ki je napisana v programskem jeziku C. Ker so slike dejansko matrike, lahko s to knjižnico izvajamo matrične operacije, kot je konvolucija in množenje matrik. Način njegove uporabe je primerljiv s programskim jezikom MATLAB [26], ki je eden izmed primarnih programov za računalniški vid in manipulacijo s slikami. SciPy [27] je znanstvena knjižnica, ki vključuje napredne analitične in statistične metode in vsebuje tudi orodja za obdelavo slik. V našem delu smo SciPy knjižnico uporabljali predvsem zaradi optimiziranih algortimov, ki jih ponuja, kot je npr. konvolucija.

3.1.3 Programski paket za strojno učenje

Pri razpoznavi znakov je potrebno vektorje značilk tudi klasificirati v posamezen razred. Knjižnica je odprtokodna razširitev namenjena uporabi v programskem jeziku Python. Vsebuje različne klasifikacijske, regresijke algoritme, kot so metoda podpornih vektorjev [12], umetne nevronske mreže [28], naivni bayesov klasifikator [28], k-najbližjih sosedov [12] in drugi. Scikit-learn knjižnica je namenjena za uporabo programskega jezika Python [21] ter numeričnih in znanstvenih knjižnic NumPy [25] in SciPy [27]. Z uporabo knjižnice scikit-learn [29] smo lahko analizirali različne metode razpoznave znakov v slikah naravnih scen in evaluirali njihovo točnost.

3.1.4 Programski paket za izris grafov

Za grafično predstavitev analiz ter rezultatov smo uporabljali knjižnico Matplotlib [30], ki je namenjena za uporabo s programskim jezikom Python [21] in razširitvenim paketom NumPy [25].

Poglavje 4

Zbirke slik teksta naravnih scen

Pri evalvaciji algoritmov smo uporabili več različnih javno dostopnih naborov podatkov. Za primerjavo rezultatov z drugimi metodami smo uporabili standardne zbirke slik teksta naravnih scen, kot je zbirka ICDAR 2003 [4], Chars74K [31] in CVL OCR DB [5]. Prednost standardiziranih podatkovnih zbirk slik je ta, da so lahko različni algoritmi objektivno evalvirani in primerjani med seboj z enakimi realnimi podatki. Večinoma se algoritme testira na podatkih, ki so prirejeni določenemu problemu, ta problem rešujejo javno dostopne zbirke. Pri evalvaciji metod je zelo pomembno, da so te evalvirane na enakih podatkih. Generirali smo tudi lastno sintetično podatkovno zbirko slik teksta z namenom izboljšanja učne množice. Pri evalvaciji smo se osredotočili na razpoznavo znakov (A-Z, a-z), ki jih sestavlja 52 razredov, v primeru invariantnosti velikosti znaka pa samo 26 razredov. Zaradi majhne reprezentativne množice slik števil v podatkovnih zbirkah slik smo pri evalvaciji izolirali števila. Pri vključitvi števil v evalvacijo prihaja do večjih napak, saj je lahko število 1 razpoznano za mali tiskani znak "l" ali veliki tistkani znak "I", število 0 je lahko rapzpoznano za "o" ali "O", število 2 je v nekaterih primerih lahko razpoznano kot "z" ali "Z", števila 3 in 8 sta lahko razpoznani za "B", število 5 je lahko napačno detektirano za znak "s" ali "S". Zaradi teh potencialnih napačnih detekcij smo zmanjšali klasifikacijski problem iz 62 razredov na 52

razredov, saj vsebuje samo črke.

4.1 ICDAR 2003

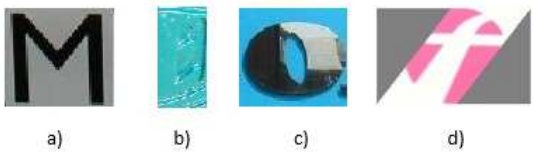
Mednarodna konferenca za analizo in razpoznavo dokumentov ICDAR [4] (angl. International Conference on Document Analysis and Recognition) je ogranizirana vsake dve leti v drugem mestu. Cilj konference je predstavitev najsodobnejših metod (angl. state-of-the-art) na področju razpoznave znakov in simbolov v slikah naravnih scen, razpoznave lastnoročne pisave, analize dokumentov, analize zgodovinskih dokumentov ter ocenjevanje uspešnosti pridobljenih rezultatov v vsaki izmed kategorij tekmovanja. Vsaka izmed kategorij je ločena na 3 segmente, detekcija teksta in segmentacija znakov, razpoznava znakov ter razpoznava besed.

Podatkovna zbirka slik ICDAR 2003 vsebuje tri različne kategorije oz. podzbirke, ki so razdeljene na vzorčno zbirko, ki vsebuje približno 850 znakov, testno zbirko slik, ki vsebuje 5400 znakov in učno zbirko slik, ki vsebuje 6100 znakov. Na učni zbirki slik smo naučili klasifikator ter nato evalvirali metode razpoznave znakov na testni zbirki slik. Vsaka podzbirka vsebuje slike znakov naravnih scen, ki so zajete v različnih pogojih. Znaki zbirke so lahko zamaknjeni, nagnjeni, zamegljeni, neenakomerno osvetljeni, ozadje slik lahko vsebuje različne barve in šume. V posamezni podzbirki se nahaja tudi XML datoteka, kjer so anotirane slike, torej vsaki sliki pripada določen znak, ki je definiran v XML elementu 4.1.

Kot je razvidno na sliki 4.2 zbirka slik teksta naravnih scen ICDAR 2003 [4] lahko vsebuje razločne znake, popolnoma nerazpoznavne znake, neenakomerno osvetljene in med drugim tudi zamaknjene znake.

```
<image file="char/3/222.jpg" tag="r" />
<image file="char/3/223.jpg" tag="d" />
<image file="char/3/224.jpg" tag="D" />
```

Slika 4.1: Primer predstavitve posameznih slik znakov v XML datoteki char.xml podatkovne zbirke ICDAR 2003 [4].

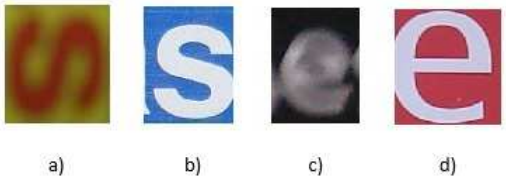


Slika 4.2: Primeri slik znakov podatkovne zbirke ICDAR 2003 [4].

4.2 Chars74K

Problem razpoznave znakov v slikah naravnih scen je evaliviran tudi na podatkovni zbirki slik Chars74K [31]. Podatkovna zbirka slik Chars74K ponuja poleg angleškega jezika tudi nabor slik znakov indijskega jezika Kannada. V verziji zbirke angleškega jezika se nahajajo tri podzbirke slik teksta in sicer sintetična zbirka slik teksta, lastnoročna zbirka slik teksta (3400 slik), ki je generirana s pomočjo računalniške tablice ter zbirka slik teksta naravnih scen, ki vsebuje dve podkategoriji in sicer slabe slike ter kvalitetnejše slike 4.3. Z vsemi tremi podzbirkami vred, šteje podatkovna zbirka slik teksta čez 74000 slik, kar nakazuje tudi ime podatkovne zbirke 74K.

V podzbirki zbirk slik teksta naravnih scen se v kategoriji slabih slik nahaja 4800 slik, v kategoriji dobrih slik pa 7700 slik znakov. Obe kategoriji vsebujeta 62 razredov (0-9, A-Z, a-z) oz. 62 direktorijev. V vsakemu direktoriju se nahaja posamezen razred znakov, zato smo preimenovali direktorije, tako da so pravilno označeni, kot je razvidno na sliki 4.4. Posamezne slike znakov smo lahko nato pravilno klasificirali glede na pripadajoč direktorij oz. njegovo zadnjo črko ločeno z znakom "_".



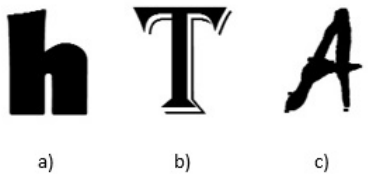
Slika 4.3: Primeri znakov slabih znakov a) in c), ki predstavljata znaka "S" in "e" ter razločnih znakov b) in d), ki prav tako predstavljata znaka "S" in "e", v zbirki slik teksta Chars74K [31].

```
dir_22_char_M
dir_35_char_Z
dir_45_char_j
```

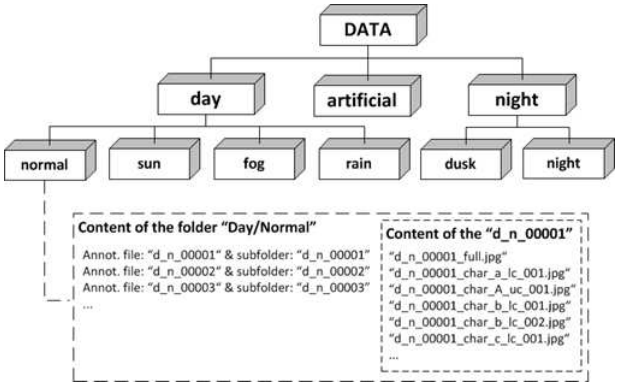
Slika 4.4: Preimenovani direktoriji posameznih razredov zbirke slik teksta Chars74K [31].

Za vsako sliko znaka se v zbirki slik teksta nahaja tudi ujemajoča se binarna segmentacijska maska znaka. V podzbirki lastnoročnih slik teksta se nahajao tudi trajektorije pisala, torej je lahko ta zbirka evalvirana na lastnoročnih znakovno razpoznavnih metodah.

Podatkovna zbirka vsebuje tudi podzbirko znakov, v kateri se nahaja 63000 sintetičnih znakov različnih tipov pisav. Ti sintetični znaki ne poustvarjajo naravnih pogojev, kot je razvidno na sliki 4.5. Znaki so binarizirani in popolnoma razločni, z namenom generiranja velike učne množice.



Slika 4.5: Primeri sintetičnih slik znakov podatkovne zbirke Chars74K [31].

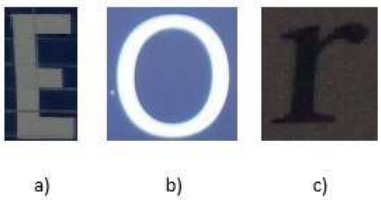


Slika 4.6: Arhitektura podatkovne zbirke slik teksta naravnih scen CVL OCR DB [5].

4.3 CVL OCR DB

Metode razpoznave znakov v slikah naravnih scen smo evalvirali tudi na podatkovni zbirki slik teksta v naravnih scenah CVL OCR DB [5]. Slednja zbirka je javno dostopna in anotirana, vsak uporabnik lahko prispeva k razširitvi zbirke, katera bi sčasoma organsko rastla. Podatkovna zbirka vsebuje 7000 slik znakov. Prednost zbirke je tudi to, da so bili v generiranje zbirke vključeni uporabniki, ki niso imeli predhodnega znanja o optični razpoznavi znakov, torej so zajemi slike na popolnoma objektivni ravni, kjer te niso bile prirejene za strokovnjaka računalniškega vida. Slike v zbirki so zajete pod različnimi vremenskimi in svetlobnimi pogoji. Slike so bile zajete s trgovin, prometnih znakov, reklamnih panojev. Vse slike v bazi vsebujejo informacijo o lokaciji na celotni sliki pripadajočih znakov. V vsakem direktoriju je nabor znakov, ki pripadajo določeni sliki. Posamezni znaki so segmentirani in izrezani iz celotne slike, vsakemu direktoriju pripada XML datoteka, ki opisuje posamezno sliko s pripadajočim znakom.

Organiziranost datotek in direktorijev je razvidna na sliki 4.6, kjer vsako izmed vozlišč predstavlja svoj direktorij. V vsakem direktoriju se nahaja množica slik teksta. Podatkovna zbirka je razdeljena v več kategorij in podka-



Slika 4.7: Primeri slik znakov podatkovne zbirke CVL OCR DB [5]. Znak a) je bil zajet pod umetno svetlobo, znak b) pri naravni svetlobi in znak c) ponoči.

```
d_f_00325_char_C_uc_001.jpg,
d_f_00325_char_C_uc_002.jpg,
d_f_00325_char_a_lc_001.jpg,
```

Slika 4.8: Poimenovanje datotek v podatkovni zbirki slik teksta CVL OCR DB [5].

tegorij, ki predstavljajo vremenske in svetlobne pogoje. Pri evalvaciji podatkov nismo razlikovali, v kateri podkategoriji se nahaja določen znak, ampak smo uporabili skupek celotne zbirke. Vse zajete slike so razdeljene v 3 kategorije "dan", "noč", "umetno". Kategorija "dan" sovpada s slikami, ki so bile zajete podnevi, "noč" slikam, ki so bile zajete ponoči in "umetno" slikam, ki so zajete pod umetno svetlobo, kot so npr. nakupovalni centri (4.7). Kategorija "dan" je razdeljena na 4 podkategorije "normalno" (dnevna svetloba), "sonce" (močna sončna svetloba), "megla" in "dež". Prav tako, je razdeljena kategorija "noč" na 2 podkategoriji in sicer "mrak" in "noč". Znak posamezne slike smo labelizirali s pomočjo razčlenitve celotnega imena slike.

Vsako končno vozlišče ustreza lokaciji slike v posameznemu direktoriju, kot je razvidno na sliki 4.6. Datoteke slik so poimenovane po ustrezni lokaciji v določenemu direktoriju. Ime datoteke se začne s predpono, ki predstavlja njeno lokacijo v datočeni strukturi, npr. "d_f" pomeni, da se slika nahaja v kategoriji

Slika 4.9: Primeri slik znakov sintetične podatkovne zbirke slik teksta.

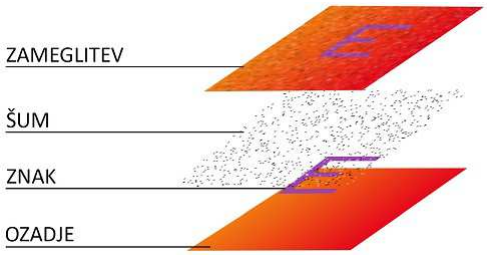
dan in njeni podkategoriji megla 4.8. Tej informaciji sledi 5-mestna številka, ki predstavja zaporedno številko celotne slike v tem direktoriju. Zaporedni številki znaka sledi konstantna beseda "char" (slov. znak), tej pa sledi dejanski znak in definiranje velikosti znaka, "up" (angl. upper char) za veliki znak in "lc" (angl. lower char) za mali znak. Vsak direktorij lahko vključuje tudi celotno sliko teksta, kjer so vsebovane celotne beseden oz. nesegmentirani znaki, tem datotekam smo se izognili, saj ne doprinesejo k razpoznavi posameznega znaka.

4.4 Sintetična podatkovna zbirka slik

Zaradi potreb po robustni učni množici smo implementirali orodje za generiranje sintetičnih znakov, ki simulirajo realne pogoje. Pri implementaciji sintetične zbirke slik znakov, smo se osredotočili na poustvarjanje vremenskih pogojev in naključnosti znakov. Slike sintetične podatkovne zbirke so enakih velikosti, torej uniformne širine in višine. Znaki so prikazani v celoti, torej noben znak ni odrezan ali prikazan samo polovično. Velikost sintetične slike je poljubna, saj se tej prilagaja tudi velikost znaka.

Pri generiranju sintetičnih znakov smo uporabili paket 2000 različnih zastonjskih pisav. Želeli smo uporabiti karseda različne vrste pisav (kot je razvidno na sliki 4.10), ki so prisotne tudi v slikah naravnih scen. Z večjo raznolikostjo tipa pisav lahko bolje naučimo učni model oz. klasifikator, ki bo za vhod prejel podoben ali enak tip pisave .

Slika 4.10: Prikaz raznolikosti pisav v sintetični zbirki. Vsi znaki predstavljajo črko "k".



Slika 4.11: Prikaz nivojev generiranja posameznega sintetičnega znaka.

Generator sintetičnih znakov smo razdelili na 4 nivoje:

- Generiranje ozadja slike
- Generiranje znaka
- Generiranje šuma
- Generiranje efekta

Vsak nivo ima svoj lasten direktorij, kjer se nahajajo korenspondenče slike. Struktura direktorijev sintetične baze je razdeljena na ozadje (angl. background), znak (angl. character), šum (angl. noise), kot je razvidno na sliki 4.11.

Ozadje znaka smo generirali, tako da ta vsebuje uniformno ozadje naključne barve oz. enobarvno ozadje ali pa je to definirano z naključnin dvobarvnim gradientom, s katerim smo želeli poustvariti neenakomerno osvetljenost znaka. Sliko znaka smo generirali iz množice 2000 različnih tipov pisav. Znakom smo določili naključne parametre kot so lokacija na slki, rotacija, nagnjenost in velikost znaka. Vsi parametri so dinamični oz. relativni glede na velikost celotne

```
calibrii_$_A
```

Slika 4.12: Poimenovanje datotek v sintetični podatkovni zbirki slik teksta.

slike. Velikost samega znaka je podana naključno v nekem razponu, vendar znak nikoli ne preseže velikosti celotne slike. Tretji nivo predstavlja šum, katerega smo generirali na dva načina in sicer s šumom sol in poper (angl. salt and pepper), kot je razvidno na sliki 4.11 ter z naključno postavitvijo elementov na sliko. V tretjem koraku smo združili skupaj tri nivoje in na končno sliko aplicirali naključen efekt. Kot je razvidno na sliki 4.11 smo na zadnjem nivoju aplicirali zameglitev celotne slike, s čimer smo simulirali nekvalitetno fotografijo slike naravnih scen. Poleg zameglitve slike smo uporabili, še ostale efekte kot je sprememba kontrasta slike (kontrast smo razdelili v štiri območja na sliki) ter poostritev slike.

Vsako datoteko smo pomenovali tako, da ta vsebuje ime tipa pisave in znak, ki ponazarja črko, ločeno s simobolom "$", kot je razvidno na sliki 4.12.

S pomočjo orodja za generiranje sintetične podatkovne zbirke lahko generiramo poljubno število znakov, mi smo za naše potrebe generirali množico slik v velikosti 13000 znakov, torej 6500 znakov malih črk ter 6500 znakov velikih črk.

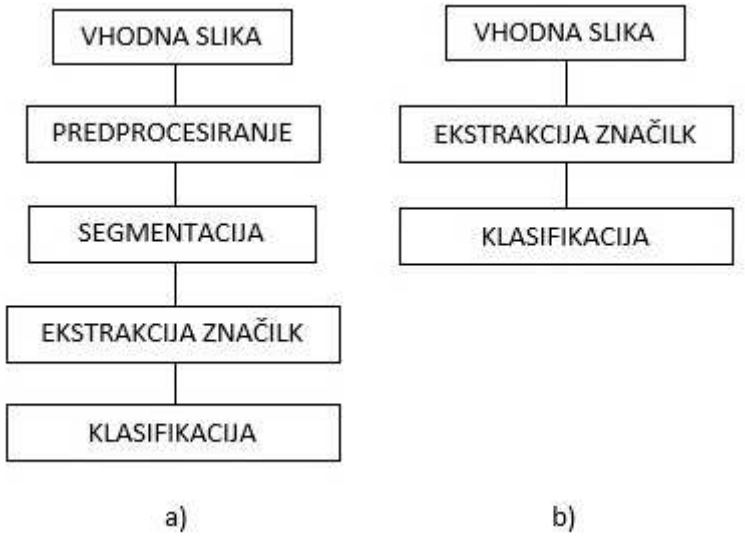
Poglavje 5

Uporabljene metode

5.1 Uvod

Pri implementaciji metod smo se osredotočali na metode, ki ne uporabljajo predprocesiranja, saj te prinesejo boljše klasifikacijske rezultate, prav tako pa te metode niso tako računsko potratne, saj preskočijo nekaj korakov in so primerne za realnočasovno razpoznavo znakov. Na sliki 5.1 je prikazana primerjava med klasičnim in modernim OCR sistemom. Slednjega smo uporabili pri implementaciji naših algoritmov.

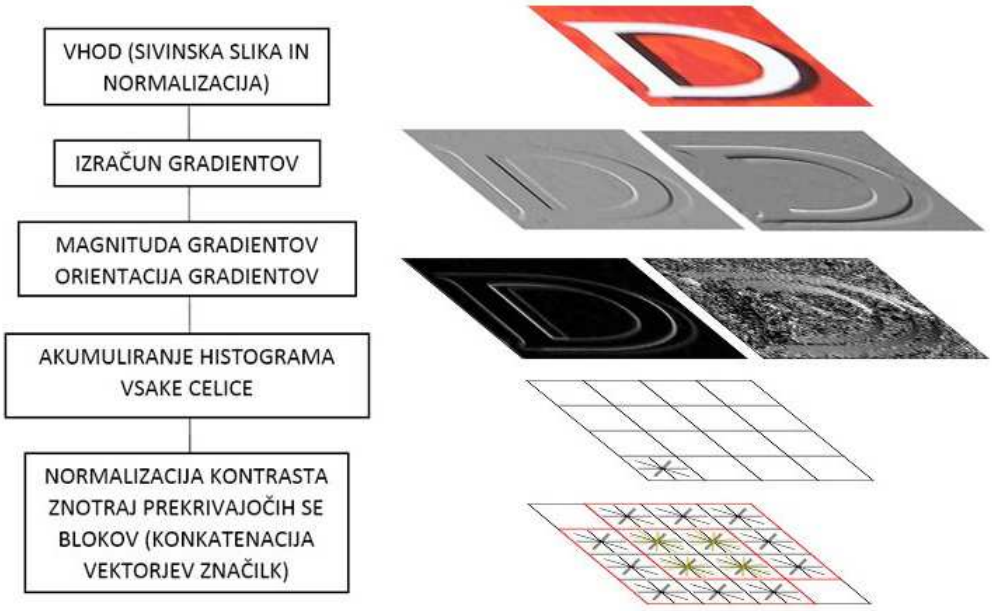
Na prvem nivoju OCR sistem prejme kot vhod sliko, ta je bila zaradi narave naših algoritmov za ekstrakcijo značilk normalizirana na velikost 32×32. Vhodno sliko smo najprej pretvorili v matriko, ki vsebuje 3 barvne kanale, pri branju slik smo uporabili odprtokodno knjižnico OpenCV ter jo pretvorili v sivinsko sliko. V drugem koraku smo iz vhodne slike ekstrahirali relevantne podatke s pomočjo različnih algoritmov. V zadnjem koraku smo algoritme za pridobivanje značilk iz znakov evalvirali na različnih klasifikacijskih algoritmih.



Slika 5.1: Diagram a) prikazuje klasični OCR sistem, diagram b) prikazuje novejši OCR sistem.

5.2 Algoritmi pridobivanja značilk

Na nivoju ekstrakcije značilk smo implementirali metode, ki temeljijo na histogramu orientiranih gradientov ter pri tem optimizirali parametre posameznih algoritmov. Na področju razpoznave vzorcev in procesiranja slik ekstrakcija značilk izhaja iz začetnega niza izmerjenih podatkov in nadgrajuje pridobljene vrednosti, ki so informativne in ne-redundantne. Ekstrakcija značilk je povezana z redukcijo dimenzionalnosti. V primeru razpoznave znakov lahko to enačimo s tem, da bi namesto ekstrakcije značilk uporabili kar celotno matriko slike, ki bi jo nato posredovali klasifikatorju, vendar bi ta vektor vseboval redundantne in ne-relevantne podatke. V primeru ekstrahiranih značilk pričakujemo, da vsebujejo relevantne podatke slike znaka. Torej lahko namesto celotne slike uporabimo reducirano predstavitev slike, ki jo ponazorimo z vektorjem značilk. Izbira relevantnih nizov značilk je poglavitni korak, za natančnejšo klasifikacijo slik znakov.



Slika 5.2: Vizualizacija HOG metode.

5.2.1 HOG

Metoda histogram orientiranih gradientov oz. HOG [17] je bila primarno namenjena za detekcijo objektov, z modificiranjem originalne implementacije pa lahko HOG modificiramo, tako da je ta primeren tudi za razpoznavo znakov v slikah naravnih scen. HOG vsebuje značilnosti algortima SIFT [20], vendar je slednji patentiran in ni na voljo za komericalno uporabo brez plačila.

Deskriptor značilk HOG je izračunan na gosti površini prekrivajočih se mrež. Zasnovan je tako, da je robusten na majhne spremembe lokacij obrisov na sliki in njihovih smeri ter hkrati robusten na velike spremembe osvetlitve in barve na sliki. Medtem ostaja diskriminatoren za celotno vizualno obliko. Za pridobitev dobrih klasifikacijskih rezultatov razpoznave s HOG metodo je primerna kombinacija gradientov brez glajenja, interpoliranje histogramov, zmerno grobo prostorsko gručenje (angl. binning) normalizacija blokov in uporaba prekrivajočih se blokov, kot je razvidno na sliki 5.2.

Vhod

HOG prejme kot vhod celotno sliko oz. njeno matriko. Slednja je bila pretvorjena v sivinsko sliko za doseganje boljših rezultatov. Spremenili smo ji tudi višino in širino oz. smo jo normalizirali na velikost 32×32. Slika je bila v prvem koraku normalizirana, s čimer se odpravi vpliv neenakomerne osvetlitve slike. Pri normalizaciji slike kompresiramo gama kanal (angl. Gamma Compression/Power Law) in barvo, bodisi z logaritmiranjem vsakega barvnega kanala slike ali z izračunom kvadratnega korena celotne matrike slike. Matrika B je kvadratni koren matrike A, če je produkt matrik B in B enak matriki A (5.1). Jakost teksture slike je ponavadi proporcionalna z lokalno osvetlitvijo površine, torej kompresija game omogoči reduciranje efektov lokalnega senčenja in svetlobne spremembe v sliki.

$$B = \sqrt{A}, B \times B = A \tag{5.1}$$

Kompresija game, korekcija game, gama kodiranje ali preprosto gama, je nelinearna operacija, ki kodira in dekodira svetilnost slike. Korekcija game je definirana z zakonom moči (5.2) (angl. Power Law), kjer je A konstanta, V_{out} in V_{in} sta ne-negativni realni vrednosti. A je ponavadi 1, v našem primeru je A enako 1/2, kar pomeni kvadratni koren matrike. Kvadratni koren torej porazdeli vrednosti preko celotne slike .

$$V_{out} = AV_{in}^{\gamma} \tag{5.2}$$

Gradient slike

Predhodno je na področju procesiranja slik veljalo, da naj bi se vsako digitalno zajeto sliko predprocesiralo z gaussovim filtrom, saj naj bi z glajenjem slike dosegali boljše rezultate. Vendar se v primeru HOG izkaže, da za pridobitev gradienta slike predhodno glajenje slike ni potrebno. S povečanjem glajenja se celo zmanjša klasifikacijska točnost. Gradienti [17] se pogosto uporabljajo pri detekciji robov v slikah, saj so spremembe gradientov najpogosteje vidne

Slika 5.3: Postopek normalizacije slike. Slika a) prikazuje originalno sliko, b) prikazuje sivinsko sliko, c) primer prikazuje normalizirano sliko.

pri robovih objektov. Slikovni elementi z visokimi vrednostmi gradienta so potencialni kandidati za predstavitev robov. Slikovni elementi z največjimi vrednostmi gradienta v smeri gradienta postanejo robni slikovni elementi. Te robove lahko nato povežemo pravokotno na smer gradienta. Algoritem, ki uporablja gradiente za detekcijo robov je Cannyjev detektor robov [32].

Gradient slike je usmerjena sprememba intenzivnosti ali barve v sliki. Z odvodom slike ponavadi želimo pridobiti informacije iz nje. Informacija, ki jo dobimo iz gradienta je ta, kako močno se slika spreminja po x ali y osi v sivinski lestvici. V drugem koraku torej izračunamo gradiente slike prvega reda, s katerim zajamemo silhuete in obrise ter informacijo o teksturi slike. Z gradienti prav tako povečamo robustnost algoritma na razlike v osvetlitvi. Na splošno to pomeni, da z gradientom izračunamo spremembo med dvema slikovnima elementoma. Gradient slike je definiran s sledečo formulo (5.3):

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \tag{5.3}$$

Parcialni odvod slike je generiran iz originalne slike, navadno s konvolucijo filtra oz. jedra čez sliko. Pri tem lahko uporabimo različna jedra. Za generiranje gradienta s konvolucijo (5.4) lahko uporabimo preprosto 1-dimenzionalno jedro, kot je razvidno na sliki 5.4 ali pa 2-dimenzionalni Sobel operator [33] oz. matriko velikost 3×3 (seštevek njenih elementov je enak 0), kot je razvidno na sliki 5.5.

$$\frac{\partial I}{\partial x} = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \otimes I \quad , \quad \frac{\partial I}{\partial y} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \otimes I \tag{5.4}$$

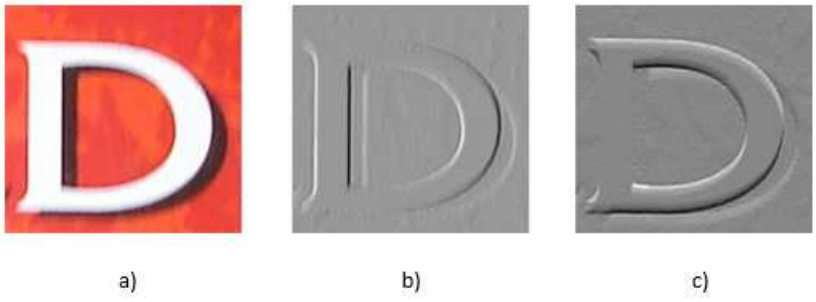
Pri konvoluciji (5.4) je pomembno, kako procesiramo podatke na robu matrike. Tu obstajajo različni načini, kjer lahko rob matrike ignoriramo, lahko ga podvojimo, zrcalimo ali vzamemo najbližje vrednosti.

$$\begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

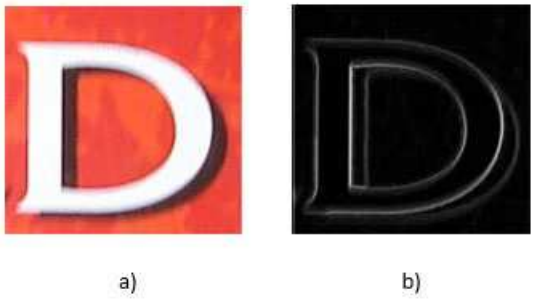
Slika 5.4: Primer 1-dimenzionalne konvolucijske matrike velikosti 1×3 ter 3×1 za izračun gradienta po x osi (leva matrika) in y osi (desna matrika). Seštevek elementov je enak 0.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Slika 5.5: Primer 2-dimenzionalnega Sobelovega operatorja oz. matrike velikosti 3×3 za izračun gradienta po x osi (leva matrika) in y osi (desna matrika). Seštevek elementov je enak 0.



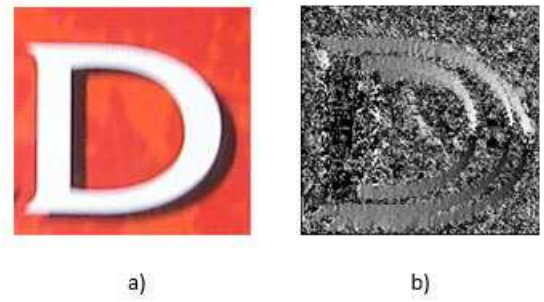
Slika 5.6: Slika a) prikazuje originalno sliko, slika b) prikazuje gradient slike po x osi, slika c) prikazuje gradient slike po y osi.



Slika 5.7: Slika a) prikazuje originalno sliko, slika b) prikazuje magnitudo gradienta slike levo od nje.

Magnituda gradientov

Z gradientom slike lahko pridobimo tudi informacijo o magnitudi gradienta 5.7. Gradient nam pove, kako močno se slika spreminja v smeri x ali y, magnituda gradienta pa, kako hitro se slika spreminja. Z magnitudo zanamo amplitudo robov, kjer slikovni elementi nenadoma spremenijo sivinski nivo. Magnitudo gradienta izračunamo kot koren seštevka kvadratov gradienta po x osi in gradienta po y osi (5.5). Magnituda gradienta se pri metodi HOG aplicira pri grajenju histograma, saj magnitudo uporabimo za definiranje uteženega histograma.



Slika 5.8: Slika a) prikazuje originalno sliko, slika b) prikazuje orientacijo gradienta slike levo od nje.

$$\|\nabla I\| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 \left(\frac{\partial I}{\partial y}\right)^2} \tag{5.5}$$

Orientacija gradientov

Poleg magnitude nam gradient poda tudi informacijo o njegovi smeri. Orientacija gradienta 5.8 nam pove, v kateri smeri se slika najbolj spreminja. Orientacijo gradienta izračunamo za gručenje magnitud v histogram, kjer z razponom orientacije definiramo, katere soležne vrednosti magnitud pripadajo histogramu.

Rezultat formule (5.6) nam vrne vrednosti radianov med $-\pi$ in π oz. $-180°$ in $180°$. Za potrebe metode HOG oz. za pridobitev najboljših klasifikacijskih rezultatov moramo orientacije pretvoriti iz radianov v stopinje ter reducirati njihov razpon na $0°$ do $180°$, saj metoda HOG bolje deluje z ne-negativnimi orientacijami. Orientacija gradienta je podana s sledečo formulo (5.6):

$$\theta = tan^{-1}\left(\frac{\partial I}{\partial x} / \frac{\partial I}{\partial y}\right) \tag{5.6}$$

$$\Big[0, 20, ..., 160, 180\Big]$$

Slika 5.9: Razpon intervala histograma metode HOG.

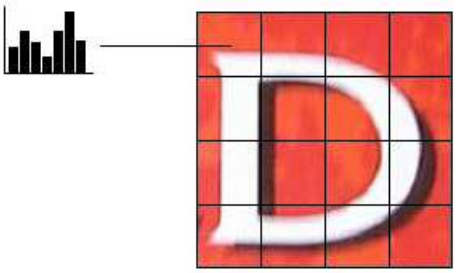
Celice slike

V tretjem koraku metoda HOG razdeli celotno sliko na posamezne celice. Vnaprej moramo definirati, koliko celic naj bi vsebovala slika. Ker smo slike predhodno normalizirali na velikost 32×32 slikovnih elementov, so imele vse slike enako število celic. Celice poimenujemo tudi prostorske regije, saj na podlagi njih izračunamo 1-dimenzionalni histogram slike. V našem primeru smo sliko širine in višine 32 slikovnih elementov razdelili na celice v velikosti 8×8 slikovnih elementov, torej smo imeli 4 celice po višini in širini oziroma 16 celic v celotni sliki, kot je razvidno na sliki 5.10.

Izgradnja histograma

Histogram je grafična predstavitev porazdelitve numeričnih podatkov in obenem ocena verjetnostne porazdelitve neprekinjene spremenljivke. Za izgradnjo histograma moramo najprej zgručiti (angl. binning) vrednosti, kar pomeni razdeliti celoten razpon vrednosti v niz intervalov in nato prešteti, kolikokrat določena vrednost pripada določenemu intervalu. Gruče oz. stolpci histograma so običajno navedeni kot zaporedni, ne prekrivajoči se intervali spremenljivk. Stolpci morajo biti med seboj sosednji in enakih velikosti po širini. Sosednji stolpci so med seboj strnjeni oz. brez razmakov, kar pomeni, da je originalna spremenljivka neprekinjena. Vertikalna os histograma predstavlja pogostost neke vrednosti v določenem intervalu (angl. bin), torej število primerov na enoto spremenljivke na horizontalni osi. Histogram, s katerim prikažemo relativne frekvence je lahko tudi normaliziran prikazuje delež primerov, ki pripadajo vsaki izmed kategorij, kjer je vsota višin intervalov enaka 1 [28].

Kot smo opisali prej, smo histograme zgradili na podlagi vrednosti vsake celice, kjer smo za vsako celico akumulirali histogram. Histogram metode HOG je razdeljen na 9 stolpcev oz. na interval med $0°$ in $180°$, kjer je širina



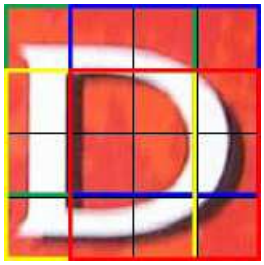
Slika 5.10: Razdelitev slike na posamezne celice v velikost 8×8 slikovnih elementov in izračun histograma vsake celice.

vsakega stolpca enaka 20° 5.9. Za izračun histograma vsake celice 5.10 smo se "sprehodili" čez vsak slikovni element te celice ter preverili v matriki orientacije gradienta v kateri interval spada določena orientacijska vrednost. V kolikor je bila orientacija gradienta na lokaciji x, y 10°, smo nato iz matrike magnitude gradienta na lokaciji x, y dodali soležečo vrednost v pravilni interval. Taki izgradnji histograma pravimo tudi uteženi histogram, saj vsaka enota pogostosti nima vrednosti 1, ampak vrednost, ki jo vsebuje matrika magnitude gradienta, kot je razvidno v formuli (5.7). Magnituda gradienta dejansko "glasuje" v posamezne stolpce histograma glede na soležeče vrednosti orientacij gradienta, ki pripadajo določenemu intervalu.

$$H(\theta) \leftarrow H(\theta) + Mag_{x,y} \tag{5.7}$$

Bloki slike

V četrtem koraku metode HOG smo sliko razdelili na več, med seboj prekrivajočih se blokov, ki vsebujejo lokalno skupino celic. Bloki so enakih velikosti 3×3 celice, kot je razvidno na sliki 5.11. Velikosti blokov so lahko poljubne, vendar nam je omenjena velikost prinesla najboljše klasifikacijske rezultate za slike velikosti 32×32 slikovnih elementov. Prekrivajoči se bloki so med seboj zamaknjeni za eno celico. Razvidno je tudi, da notranje celice slike prispevajo večim komponentam končnega vektorja značilk. Zaradi optimizacijskih razlo-



Slika 5.11: Razdelitev slike na posamezne bloke velikosti 3×3 celice. Bloki so prikazani z zeleno, modro, rdečo in rumeno barvo.

gov so histogrami celic izračunani vnaprej, torej jih v segmentu generiranja blokov ekstrahiramo iz pravilno indeksiranih celic. Vsak blok ima definiran seznam indeksov celic, ki jih vsebuje. Histograme teh celic nato pridobimo iz podatkovne strukture, v kateri je bil histogram shranjen.

Normalizacija in konkatenacija

V zadnjem koraku izračunamo normalizacijo histogramov, ki vsebuje lokalne skupine celic ter normalizira njihov odziv pred konkatenacijo. Z normalizacijo izboljšamo invariantnost algoritma na osvetljevanje, senčenje in kontrast robov. Gruče lokalnih celic oz. histogramov najprej akumuliramo v posamezen blok, kjer pridobimo vektor značilk, celoten vektor, ki je del bloka, nato normaliziramo s formulo (5.8). Navadno se posamezna celica deli med več blokov, kot je prikazano na sliki 5.11, vendar je normalizacija histogramov celic vezana na posamezen blok, torej neka celica, ki pripada dvema blokoma, prispeva drugačne vrednosti v oba. Histogram oz. posamezna celica se v končnem vektorju značilk pojavi večkrat, vendar to ni redundatno, saj prispeva k boljši klasifikacijski točnosti. Normaliziran vektor značilk posameznega bloka nato konkateniramo oz. združimo v 1-dimenzionalni vektor značilk, ki je namenjen za klasifikacijo.

$$H = H/\sqrt{\left(\sum_{i=0}^{n} H_{(i)}\right)^2 + \varepsilon} \tag{5.8}$$

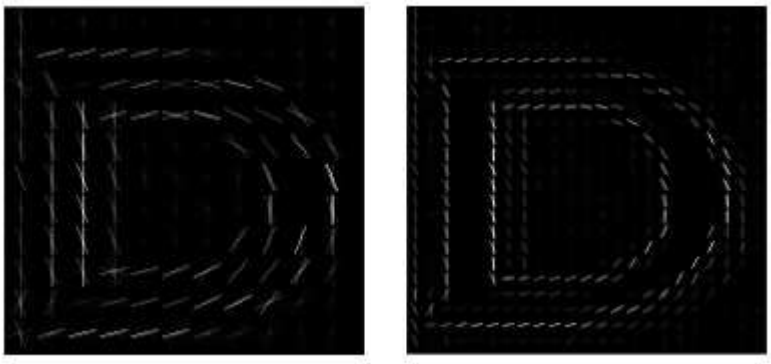
Kjer:

$$\varepsilon = 1^{-5}$$

Parametri

Algoritem HOG [17] je bil implementiran tako, da smo mu lahko spreminjali parametre, kot je število orientacij oz. stolpcev (angl. bins) v histogramu, razpon stopinj, ki naj bi jih histogram obravnaval (v našem primeru od 0° do 180°), število slikovnih enot, ki jih vsebuje posamezna celica ter število celic, ki jih vsebuje vsak blok slikovne matrike.

Na sliki 5.12 je razvidno, kako velikosti blokov in celic vplivajo na definiranje strukture znaka. Za generiranje leve slike 5.12 smo definirali bloke velikosti 1×1 celic in 16×16 slikovnih enot na celico ter 9 orientacij. Ti parametri nam podajo nedefinirano strukturo znaka, medtem ko smo za generiranje desne slike 5.12 uporabili bloke velikosti 3×3 celic in 8×8 slikovnih enot na celico ter 9 orientacij, kjer je znak tudi lažje razbran oz. bolj definiran.



Slika 5.12: Vizualizacija HOG algoritma oz. gručene orientacije gradientov z različnimi parametri. Primerjava prikazuje levo sliko z bloki velikosti 1×1 celic in 16×16 slikovnih enot na celico ter 9 orientacij in desno sliko z bloki velikosti 3×3 celic in 8×8 slikovnih enot na celico ter 9 orientacij.

5.2.2 PHOG

V splošnem poznamo piramidne predstavitve signalov ali slik, ki so predmet ponavljajočega se glajenja in podvzorčenja. Tipična piramida je navadno generirana z glajenjem slike z ustreznim filtrom glajenja in s podvzorčenjem zglajene slike, kjer je faktor pomanjšanja dvakraten za vsako os. Producirana slika je nato podvržena enakem procesu, torej je cikel ponovljen večkrat. Vsak cikel tega procesa rezultira v vedno manjši sliki s povečanim glajenjem, vendar z zmanjšano gostoto prostorskega vzorčenja (zmanjšana ločjivost slike). Ponazorjeno grafično bo celotna predstavitev izgledala kot piramida z originalno sliko na dnu in z najmanjšo sliko na vrhu [34].

Piramidni histogram orientiranih gradientov [34] je variacija originalnega HOG algoritma in izvira iz dveh virov in sicer: (1) Prezentacija slikovne piramide [35] ter (2) HOG [17] Primarno je PHOG [34] namenjen razpoznavi objektov, njegov glavni cilj je prostorska razporeditev oblik, ki bi koristila boljši razpoznavi. Metode detekcije objektov navadno uporabljajo lokalni videz (angl. local appearance) objekta in ne oblike neposredno. Vendar prestavi-

tve oblike z uporabo prostorske razporeditve robov ponavadi bolje detektirajo objekte, kot metode lokalnega videza, vendar slednje vključujejo geometrično skladnost s pomočjo Hough metode. Metoda PHOG [34] vključuje obe pozitivni lastnosti omenjenih detektorjev objektov, kjer zajema prostorske porazdelitve robov, vendar je formulirana kot vektor.

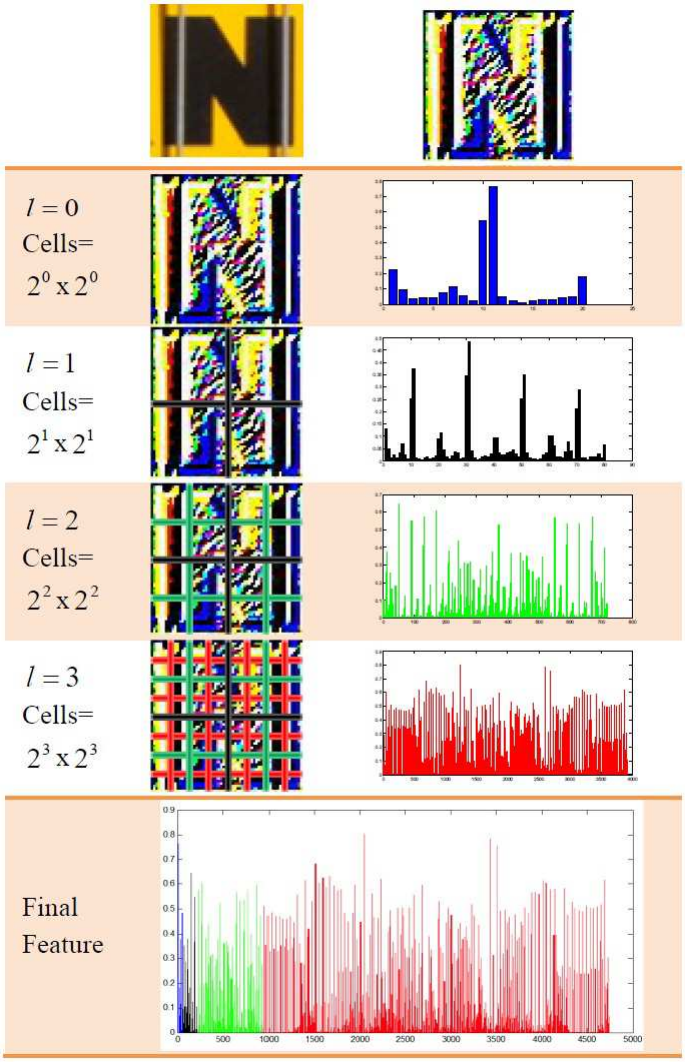
Modificiran algoritem PHOG [36], ki smo ga implementirali, temelji na ideji originalne verzije algoritma PHOG [34] in je namenjen za razpoznavo znakov v slikah naravnih scen. PHOG [36] metoda vključuje algoritem HOG v prostorski piramidi, kot je razvidno na sliki 5.13. Glavni namen PHOG je predstavitev oblike slike in njene prostorske postavitve, tako da je lahko primerjava med dvema oblikama izračunana s SVM klasifikatorjem [12].

Vhod

Algoritem PHOG [36] prejme za vhod matriko slike. Slika je pretvorjena v sivinsko sliko, kot pri algoritmu HOG [36], hkrati je bila tudi normalizirana na velikost 32×32. Normalizacija slike oz. kompresija game, je namenjena odpravljanju neenakomerne osvetlitve slike ter reduciranju lokalnega senčenja.

Gradient slike (magnituda in orientacija gradientov)

Gradient slike smo izračunali s formulo opisano v enačbi (5.3) in pri tem uporabili jedro prikazano na sliki 5.4. V prvem koraku smo izračunali magnitudo gradientov po formuli (5.5). Magnitudo smo izračunali samo enkrat, saj je ni bilo potrebno računati za vsak piramidni nivo posebej. Prav tako kot pri metodi HOG [17] smo morali v PHOG algoritmu [36] izračunati orientacijo gradientov, saj smo za vsako celico piramidnega nivoja generirali utežen histogram orientacij z magnitudami gradienta.



Slika 5.13: Prikaz PHOG histogramov na različnih piramidnih nivojih. Povzeto po "Using Pyramid of Histogram of Oriented Gradients on Natural Scene Text Recognition" [36].

Razdelitev na celice

Metoda HOG [17] razdeli sliko na več med seboj enakih celic, pri metodi PHOG [36] pa je razdelitev celic dinamična in ne statična kot v HOG [17] metodi. Piramidna predstavitev slike pomeni, da sliko vedno znova vzorčimo v različnih prostorskih predstavitvah. Ker je histogram orientiranih gradientov predstavljen kot mreža celic, se pri metodi PHOG ta mreža zgosti z vsakim naslednjim nivojem oz. korakom. Mrežo celic torej z vsakim korakom bolj zgladimo. Kot pri navadnem piramidnem sistemu tudi tu vsaka celica postane manjša z vsakim naslednjim nivojem ter izgubi resolucijo oz. število slikovnih elementov. Na najvišjem nivoju bo predstavljen prvi nivo oz mreža velikosti enega bloka, nato pa bo mreža vedno bolj zgoščena. Predstavitev mreže vsakega nivoja je definirana s sledečo formulo (5.9):

$$2^l \times 2^l \tag{5.9}$$

Kjer:

$$l = \text{piramidni nivo}$$

PHOG [36] lahko vsebuje poljubno število piramidnih nivojev, vendar smo se zaradi problema prezasičenosti in računske nezahtevnosti omejili na 4 nivojsko piramido. Kot lahko sklepamo iz formule (5.9) bo vsak naslednji piramidni nivo vseboval eksponentno več celic ter bo hkrati tudi računsko bolj zahteven, prav tako bo vektor značilk z vsakim nivojem večji. Končni vektor značilk pa bo skupek posameznih nivojev, velikost končnega vektorja značilk je predstavljena s sledečo formulo (5.10):

$$K \cdot \sum_{l \epsilon L} 4^l \tag{5.10}$$

Kjer:

K = število orientacij

L = končno število piramidnih nivojev

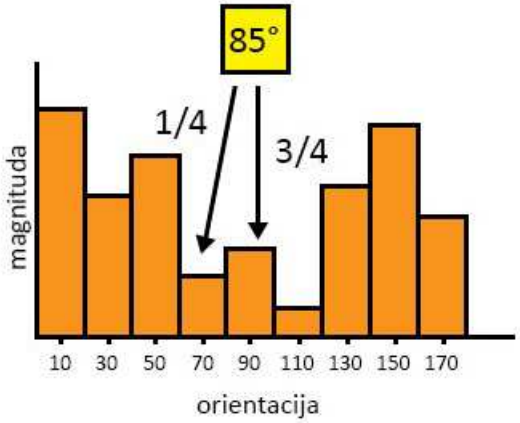
l = piramidni nivo

Grajenje histograma in bilinearna interpolacija

Utežene histograme smo izračunali iz vsake celice na vsakem nivoju. Histogram smo utežili z magnitudo gradientov, po formuli (5.7), kjer smo gručili magnitude po soležni orientaciji gradienta, ki spada v določen interval histograma na lokaciji x, y. Zaradi piramidne predstavitve histogrami na različnih piramidnih nivojih predstavljajo znak drugače. Histogrami so na višjem piramidnem nivoju gostejši in podajajo več informacij kot histogrami na nižjem nivoju.

Bilinearno interpolacijo smo uporabili pri dodeljevanju vrednosti magnitud, glede na njihovo soležno orientacijsko vrednost v posamezen orientacijski interval. Pri dodelitvi vrednosti magnitude smo upoštevali, da lahko neka magnituda prispeva tudi sosednjemu stolpcu, glede na njegovo bližino. Torej, če je bila orientacijska vrednost 96°, je ta polega tega da je prispevala utež oz. magnitudo v interval med 80° in 100°, prispevala še v bližnji interval oz. stolpce (100° in 120°), saj je vrednost 96° bližje centru desnega stolpca 110° kot centru levega stolpca s 70°. Sosednjemu stolpcu prispeva toliko kolikor je zmnožek magnitude z razliko med orientacijsko vrednostjo ter centrom njenega stolpca, ulomljeno z razliko med centroma stolpcev, kot je ponazorjeno na sliki 5.14 in z enačbo (5.12). Svojemu stolpcu prispeva zmnožek magnitude z razliko med to utežjo sosednjega stolpca in vrednostjo 1, kar je razvidno v enačbi (5.11). Bilinearna interpolacija zgladi histogram ter razporedi vrednosti magnitud enakomerno, s čimer zmanjšuje nepravilnosti.

$$H(\theta_1) \leftarrow H(\theta_1) + mag_{x,y} * (1 - \beta/\gamma)) \tag{5.11}$$



Slika 5.14: Prikaz bilinearne interpolacije med dvema stolpcema histograma, kjer je $\theta = 85°$. Razlika do središča stolpca 70 je 15°, razlika do središča stolpca 90 je 5°. Koeficienti za množenje z magnitudo so torej $5/20 = 1/4$ ter $15/20 = 3/4$.

$$H(\theta_2) \leftarrow H(\theta_2) + mag_{x,y} * (\beta/\gamma) \tag{5.12}$$

Kjer:

θ_1 = interval, kateremu pripada trenutna vrednost orientacije gradienta

θ_2 = sosednji interval, z najmanjšo razliko orientacije do centra

mag = magnituda gradienta na lokaciji x,y

β = razlika vrednosti orientacije gradienta in njenega centra intervala

γ = razlika med dvema centroma intervalov

Trilinearna interpolacija [36] naj bi nastopila na drugem piramidnem nivoju in zmanjšala artefakte histograma ter distorzije. Vendar smo se zaradi računske zahtevnosti te implementacije odločili ta korak preskočiti, saj v tem primeru PHOG [36] ne bi bil več primeren za realnočasovne aplikacije. Tudi

- če $l = 0 \leftarrow$ vrni blok velikosti 1×1 celic
- če $l = 1 \leftarrow$ vrni blok velikosti 2×2 celic
- če $l = 2 \leftarrow$ vrni blok velikosti 3×3 celic
- če $l = 3 \leftarrow$ vrni blok velikosti 7×7 celic

v nekaterih HOG implementacijah se trilinearne interpolacije navadno ne implementira zaradi omenjenega razloga.

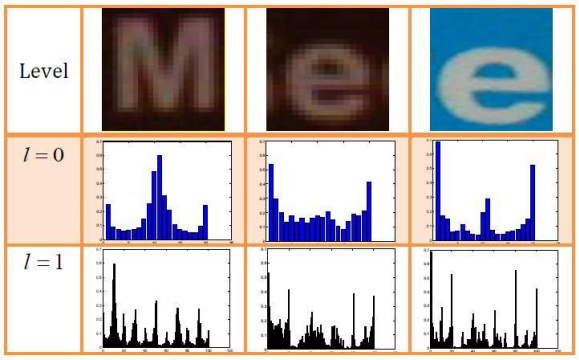
Bloki slike

V četrtem koraku metode PHOG smo sliko, prav tako kot v metodi HOG razdelili na več, med seboj prekrivajočih se blokov. Zaradi piramidnega sistema metode PHOG smo morali bloke dinamično spreminjati glede na trenutni nivo piramide.

Na prvem nivoju piramide smo imeli samo eno celico, torej je bil histogram definiran z enim blokom v velikosti 1×1. Velikost blokov je torej sorazmerna s stopnjo piramidnega nivoja, v katerem se trenutno nahajamo. Velikosti smo definirali po sledečem obrazcu (kjer l predstavlja piramidni nivo):

Normalizacija in konkatenacija

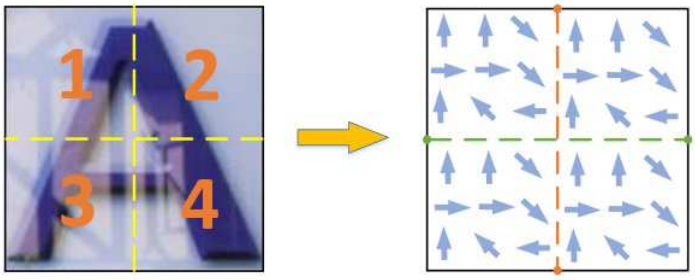
Na vsakem piramidnem nivoju smo izračunali histograme iz posameznih celic različnih velikosti ter jih razdelili v več blokov. Vektorje značilk pridobljene iz histogramov posameznega bloka smo nato normalizirali po formuli (5.8). Normalizirane vektorje značilk smo na vsakem piramidnem nivoju združili v 1-dimenzionalen vektor. V zadnjem koraku smo konkatenirali vse 4 vektorje značilk, ki so pripadali posameznemu piramidnem nivoju, v 1-dimenzionalen vektor značilk, kot je razvidno na sliki 5.13.



Slika 5.15: Prikaz PHOG alogritma na različnih znakih. Različni znaki na podobnem ozadju imajo povsem drugačne oblike histogramov, medtem ko ima enak znak na drugačnem ozadju podobno obliko histograma. Povzeto po "Using Pyramid of Histogram of Oriented Gradients on Natural Scene Text Recognition" [36].

Parametri

Implementacija metode PHOG [36] je vključevala tudi definicijo parametrov, ki jih algoritem potrebuje. Algoritem je torej kot parametre metode prejel vhodno sliko, število orientacijskih intervalov histograma, razpon stopinj orientacij gradienta ter število piramidnih nivojev. Velikost celic je dinamično definirana za vsak piramidni nivo, prav tako je število blokov na vsakem piramidnem nivoju drugačno.



Slika 5.16: Delitev vhodne slike na 4 bloke. Povzeto po "Scene Text Recognition using Co-occurrence of Histogram of Oriented Gradients" [37].

5.2.3 Co-HOG

V magistrskem delu smo implementirali različico Co-HOG [37] metode namenjene razpoznavi znakov v slikah naravnih scen. Primarno je Co-HOG [38] namenjen detekciji ljudi, vendar se izkaže tudi kot dober algoritem za razpoznavo znakov v slikah naravnih scen. Co-HOG [37] temelji na algoritmu HOG [17], ki je robusten na spremembo svetilnosti, geometrične in fotometrične spremembe, vendar je HOG le statistika usmerjenosti gradienta vsakega bloka, ki ne zajema prostorske informacije med sosednjimi slikovnimi elementi.

Co-HOG [38] predstavlja histograme, katerega gradniki so pari orientacije gradienta. Ker Co-HOG [37] zajame tudi orientacijo gradienta sosednjih slikovnih elementov, lahko prikaže več informacij ter izrazi obliko znaka bolj podrobno kot HOG, v katerem se uporablja orientacija gradienta enega slikovnega elementa. Co-HOG [37] je hitrejši od metode HOG, saj sliko razdeli na neprekrivajoče se bloke 5.16, kar je pomembno pri realnočasovni razpoznavi. Pomembno je tudi, da Co-HOG upošteva relativno lokacijo in orientacijo gradienta vsakega sosednjega slikovnega elementa, s čimer bolj natančno opiše obliko znaka. Obenem Co-HOG [37] ohranja prednosti algoritma HOG [17] kot so invariatnost na neenakomerno osvetljenost in lokalne geometrične transformacije.

Vhod

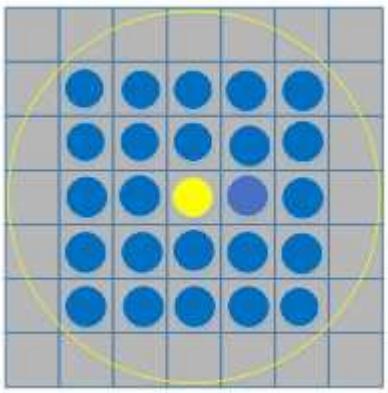
Co-HOG algoritem prejme za vhod barvno ali sivinsko sliko velikosti 32×32 slikovnih elementov. V primeru barvne slike, ki vsebuje 3 barvne kanale in sicer rdečega, modrega in zelenega smo za vsak barvni kanal izračunali magnitudo, ki je bila predstavljena kot matrika. Na podlagi te matrike smo nato izračunali vsoto vrednosti matrike in izbrali barvni kanal slike z navječjo magnitudo. Izbran barvni kanal smo nato pretvorili v sivinsko barvno lestvico ter ga normalizirali s kvadratnim korenom matrike (5.1). Enako smo storili s sivinsko sliko, le da smo v tem primeru preskočili filtriranje kanalov, saj vsebuje samo enega.

Gradient slike (magnituda in orientacija gradientov)

Gradient slike je bil izračunan s formulo opisano v enačbi (5.3), kjer je bil uporabljen Sobel operator 5.5. Implementacija metode Co-HOG tudi vključuje izračun magnitude (5.5) ter orientacije gradientov (5.6), saj za generiranje 3-D histograma potrebujemo te vrednosti. Orientacije gradientov segajo v razponu od 0° do 180°, kjer je gradient nenegativen oz. nepredznačen, orientacije so nato kvantizirane v 9 intervalov.

Grajenje blokov

Co-HOG razdeli sliko na več blokov, vendar ti niso med seboj prekrivajoči, kot pri metodi HOG [17] in PHOG [36]. Iz vsakega izmed blokov smo z orientacijo gradienta ter magnitudo gradienta izračunali so-pojavitvene matrike oz. 3-D histograme ter jih nato konkatenirali v 1-dimenzionalni vektor značilk. V našem primeru se je izkazalo, da so prinesli najboljše rezultate bloki velikosti 8×8 slikovnih elementov, kar je na vhodni sliki velikosti 32×32 slikovnih elementov pomenilo, da slednja vsebuje 16 blokov oz. je velika 4×4 bloka.

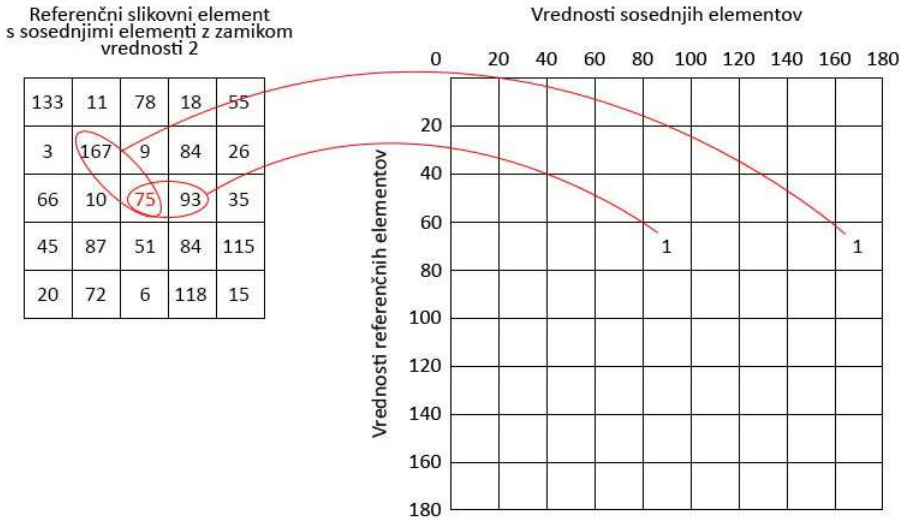


Slika 5.17: Zamik (2 slikovna elementa) s katerim izračunamo so-pojavnost orientiranih gradientov med pari slikovnih elementov. Povzeto po "Scene Text Recognition using Co-occurrence of Histogram of Oriented Gradients" [37].

Grajenje 3-D histograma in bilinearna interpolacija

Metoda Co-HOG [38] zajame prostorske informacije s štetjem frekvence so-pojavnosti orientiranih gradientov med pari slikovnih elementov, pri čemer so shranjene relativne lokacije. Relativne lokacije odraža zamik med dvema slikovnima elementoma, prikazano na sliki 5.17. Sredinski, rumeni slikovni element je trenuten element, ki ga obravnavamo, njegovi sosednji elementi so obarvani z modro barvo, sosednji elementi so oddaljeni z zamikom dveh slikovnih elementov. Vsak sosednji slikovni element (modre barve) tvori orientacijski par s sredinskim (rumenim) slikovnim elementom in ustrezno glasuje v so-pojavitveno matriko, kot je prikazano na sliki 5.19. Ker ima so-pojavitvena matrika zamik dveh slikovnih elementov ima sredinski element 24 sosednjih elementov, kot je razvidno na sliki 5.17. Co-HOG je razširitev algoritma HOG, s katerim se ujema, ko je zamik slikovnih elementov enak $(0, 0)$.

So-pojavitvena matrika ali so-pojavitvena porazdelitev je matrika, ki je definirana s sliko, porazdelitve so-pojavitvenih vrednosti pri določenem zamiku. Matematično je so-pojavitvena matrika definirana z enačbo (5.13). V splošnem lahko katerakoli matrika generira so-pojavitveno matriko, vendar je navadno

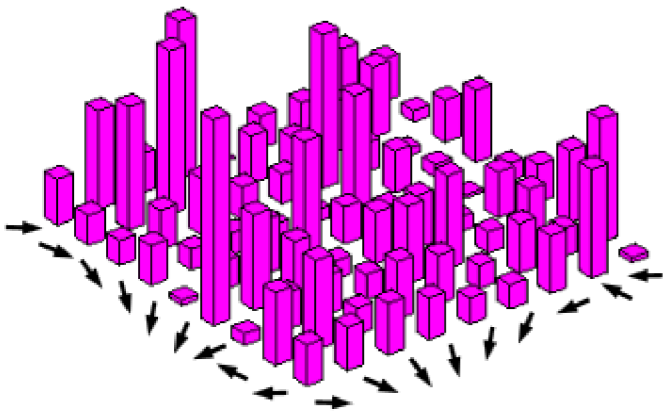


Slika 5.18: Prikaz izgradnje so-pojavitvene matrike. Leva matrika prikazuje matriko sosednjih elementov trenutno obravnavane orientacijske vrednosti. Desna matrika prikazuje so-pojavitveno matriko, kjer sta dodeljeni vrednosti frekvenc pojavitev označenih orientacij.

uporabljena pri merjenju tekstur v slikah, zato ponavadi predpostavljamo, da matrika ponazarja sliko. Zaradi parametrov x in y v enačbi (5.13) je so-pojavitveno matrika občutljiva na rotacijo.

So-pojavitvena matrika je kvadratna matrika in je definirana s številom orientacijskih intervalov oz. stolpcev (angl. bins). V našem primeru je dimenzija so-pojavitvene matrike $9 \times 9 \times n$, kjer n predstavlja višino posameznega stolpca 5.19. So-pojavitveno matriko smo generirali, tako da smo za vsak referenčni element in njegov sosednji element, vpisali število njunih pojavitev v matriko, kot je prikazano na sliki 5.18.

$$H_{x,y}(i,j) = \sum_{(p,q)\epsilon B} \begin{cases} 1 & \text{če } O(p,q) = i \ \ \& \ \ O(p+x, q+y) = j \\ 1 & \text{sicer} \end{cases} \tag{5.13}$$



Slika 5.19: 3-dimenzionalni histogram so-pojavitvene matrike Co-HOG [37] algoritma. Povzeto po "Scene Text Recognition using Co-occurrence of Histogram of Oriented Gradients" [37].

Kjer:

$H_{x,y}$ = so-pojavitvena matrika z zamikom (x,y)

O = matrika orientacije gradienta vhodne slike

B = blok slike

p, q = prostorska lokacija v matriki magnitude gradienta

i, j = vrednosti matrike orientacije gradienta

x, y = zamik po x in y oseh

Originalna Co-HOG metoda ne vključuje uteževanja so-pojavitvene matrike, temveč jo zgradi po formuli opisani v enačbi (5.13), ki sama po sebi ne izrazi razlike med močnim gradientom in šibkimi vrednostmi gradienta. V našem delu smo implementirali izboljšano metodo Co-HOG [37] za razpoznavo znakov v slikah naravnih scen in vključuje uteževanje, ki temelji na magnitudi gradienta ter bilinearno interpolacijo med dvema sosednjima orientacijskima stolpcema (angl. bin), kot je to prikazano v enačbi (5.14). V

predlagani enačbi za uteževanje in interpolacijo imajo lahko slikovni elementi z zelo majhno vrednostjo magnitude gradienta velik vpliv na uteževanje, v kolikor ima njen sosednji slikovni element visoko vrednost magnitude gradienta. V izogib temu ne obravnavamo teh slikovnih elementov, kjer ima vsaj eden izmed njiju zelo majhno vrednost magnitude gradienta.

$$\begin{aligned}
H(\theta_1,\theta_3) &\leftarrow H(\theta_1,\theta_3) + M_1 \left(1-\tfrac{\alpha-\theta_1}{\theta_2-\theta_1}\right) + M_2 \left(1-\tfrac{\beta-\theta_3}{\theta_4-\theta_3}\right)\\
H(\theta_1,\theta_4) &\leftarrow H(\theta_1,\theta_4) + M_1 \left(1-\tfrac{\alpha-\theta_1}{\theta_2-\theta_1}\right) + M_2 \left(\tfrac{\beta-\theta_3}{\theta_4-\theta_3}\right)\\
H(\theta_2,\theta_3) &\leftarrow H(\theta_2,\theta_3) + M_1 \left(\tfrac{\alpha-\theta_1}{\theta_2-\theta_1}\right) + M_2 \left(1-\tfrac{\beta-\theta_3}{\theta_4-\theta_3}\right)\\
H(\theta_2,\theta_4) &\leftarrow H(\theta_2,\theta_4) + M_1 \left(\tfrac{\alpha-\theta_1}{\theta_2-\theta_1}\right) + M_2 \left(\tfrac{\beta-\theta_3}{\theta_4-\theta_3}\right)
\end{aligned} \tag{5.14}$$

Kjer:

H = so-pojavitvena matrika pri določenem zamiku, kot je razvidno v enačbi (5.13)

M_1 = magnituda gradienta na lokaciji (p, q)

α = soležna orientacija gradienta M_1

M_2 = magnituda gradienta na lokaciji $(p + x, q + y)$

$beta$ = soležna orientacija gradienta M_2

θ_1, θ_2 = sosednja centra intervalov (angl. bin) orientacije gradienta α

θ_3, θ_4 = sosednja centra intervalov (angl. bin) orientacije gradienta β

Normalizacija

Ker pri metodi Co-HOG računamo so-pojavitveno matriko, nam ta vrne 3-dimenzionalen histogram na nivoju vsakega bloka. Pridobljen histogram moramo nato transformirati v 1-dimenzionalen vektor značilk, slednji je normaliziran s formulo opisano z enačbo L-2 norm (5.15). Deskriptor značilk celotne preučevane slike je nato konstruiran s konkatenacijo vseh normaliziranih vektorjev značilk posameznega bloka.

$$v \leftarrow v/\sqrt{\|v_k\|^2 + \varepsilon^2} \tag{5.15}$$

Kjer:

$$\varepsilon = 1^{-5}$$

Parametri

Metoda Co-HOG je kot parametre metode prejela vhodno sliko, število orientacijskih intervalov histograma, število blokov v sliki ter zamik so-pojavitvene matrike. Število orientacijskih intervalov (angl. bins) smo omejili na 9, število blokov v sliki je 16 (4×4). Zamik so-pojavitvene matrike je bil definiran s številom slikovnih elementov, ki je bil nastavljen na 2, kar pomeni, da ima referenčni slikovni element 24 sosedov, kot je razvidno na sliki 5.17. Pri interpolaciji smo v izogib neenakomernega uteževanja morali določiti prag, ki je definiral, katere sosednje pare vrednosti magnitud gradienta lahko obravnavamo v so-pojavitveni matriki. Prag za sprejemljive vrednosti magnitud gradientov smo nastavili od 0.1 do 0.5, saj ničelne vrednosti doprinesejo k neenakomerni uteženosti.

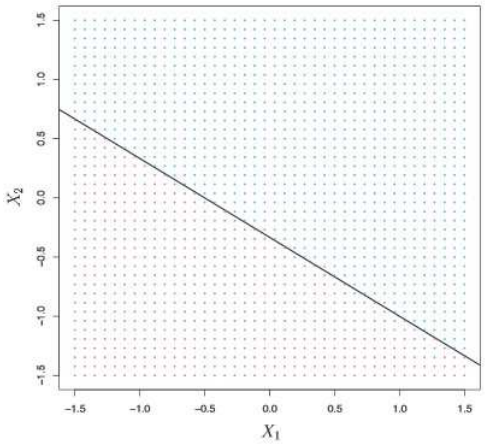
5.3 Klasifikacija

V kontekstu strojnega učenja je klasifikacija problem identifikacije kateri kategoriji pripada nek nov vzorec na podlagi učne množice. Na podlagi učne množice lahko vhodni vzorec klasificiramo binarno ali večrazredno. Več-razredna klasifikacija je razširitev binarne klasifikacije, ki razlikuje samo med dvema razredoma. V našem delu smo uporabljali večrazredne klasifikatorje, saj je vsak znak predstavljal svoj razred, v katerega smo uvrstili vhodni vzorec. Uporabljali smo nadzorovano učenje, kjer smo generirali model iz podatkov učne množice in z njim procesirali napovedi.

Za evalvacijo uporabljenih metod razpoznave znakov in različnih podatkovnih zbirk slik teksta smo uporabili več klasifikacijskih algoritmov. Za klasifikacijo smo uporabili algoritem SVM (metoda podpornih vektorjev), ANN (umetne nevronske mreže), K-NN (metoda k-najbližjih sosedov), NB (naivni Bayes). Klasifikatorje kategoriziramo v več, med seboj si podobnih načinov delovanja in sicer instančne algoritme, kamor spada K-NN, kjer se zgradi podatkovni model učne množice in se nato uporabi mera podobnosti, s katero se izračuna najboljše ujemanje. V drugo skupino spadajo Bayesovi algoritmi, kamor spada naivni Bayes, ki izrecno uporabljajo Bayesov teorem za problem kot je klasifikacija in regresija. Ena izmed večjih skupin algoritmov so tudi algoritmi umetnih nevronskih mrež, ki temeljijo na strukturi bioloških nevronskih mrež in so namenjeni za klasifikacijske in regresijske probleme. Metoda podpornih vektorjev pa spada v skupino nadzorovanih metod učenja in temelji na jedrnih tehnikah algoritmov (ang. kernel-based).

5.3.1 SVM

Temelji algoritma SVM so bili razviti s strani Vapnik et. al. 1995 [39] in je eden izmed najbolj uporabljenih algoritmov za klasifikacijo. SVM algoritme delimo na linearne in nelinearne. Metoda podpornih vektorjev je razširitev klasifikatorja podpornih vektorjev (angl. support vector classifier). Slednji te-



Slika 5.20: Slika prikazuje hiperravnino z enačbo $1+2X_1+3X_2 = 0$, ki deli dve množici točk. Modra regija množica točk ustreza enačbi $1 + 2X_1 + 3X_2 > 0$, rdeča regija pa je predstavljena z enačbo $1 + 2X_1 + 3X_2 < 0$ [12].

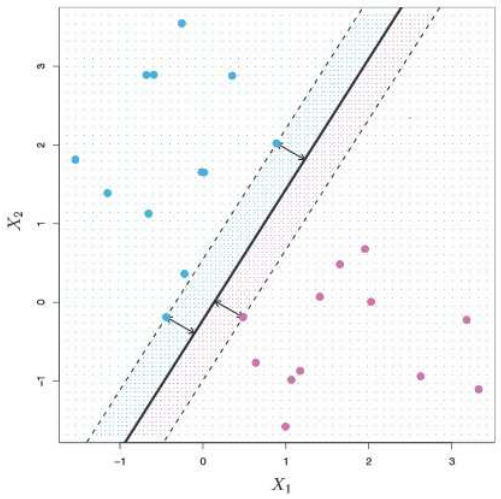
melji na maksimalnem mejnem klasifikatorju (angl. maximal margin classifier), ki je linearni klasifikator in deli ravnino na dva dela s hiperravnino in spada v skupino binarnih klasifikatorjev. V p-dimenzionalnem prostoru je hiperravnina afini podprostor dimenzije $p - 1$, torej ima 1 dimenzijo manj kot njen prostor, in je podana z enačbo (5.16). V 2-dimenzionalnem prostoru je hiperravnina 1-dimenzionalna ravna površina, v 3-dimenzionalnem prostoru pa je hipperavnina 2-dimenzionalni podprostor oz. ravnina. Lahko si predstavljamo da hiperravnina deli p-dimenzionalni prostor na dva dela, kot je prikazano na sliki 5.20.

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0 \tag{5.16}$$

Kjer:

$$X = (X_1, X_2, \ldots, X_p)^T$$

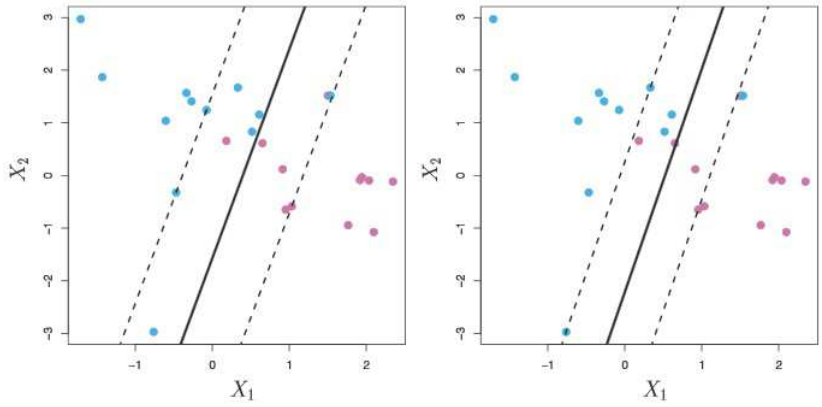
V primeru, da je možno podatke popolnoma ločiti z ravnino, potem lahko obstaja neskončno hiperravnin, saj je lahko hiperravnina malce zamaknjena ali



Slika 5.21: Slika prikazuje hiperravnino z maksimalno mejo. Meja je definirana z razliko od polne črne črte do črtkanih črt. Točke na črtkanih črtah so podporni vektorji. Modra in rdeča mreža prikazujeta odločitev klasifikatorja glede na hiperravnino [12].

rotirana, vendar bo še vedno ločevala podatke. Najti moramo torej primeren način, ki nam pove katero izmed hiperravnin izbrati. Izbrana hiperravnina bo tista, ki je najbolj oddaljena od podatkov, to pomeni da izračunamo pravokotno razdaljo od vsake učne točke do posamezne hiperravnine, kjer bo optimalna hiperravnina tista, ki ima največjo mejo do najbližjih točk, kot je razvidno na sliki 5.21. V realnih primerih podatkovne točke enega razreda "ležijo" v nasprotnem razredu, torej podatke ne moremo natančno ločiti s hiperravnino. Z uporabo mehke meje, lahko skoraj ločimo razrede, kjer lahko pride do napačne klasifikacije podatkov. Nov parameter oz. nenegativna konstanta C, ki določa število in resnost "kršitev" oz. napak, ki smo jih pripravljeni sprejeti. V kolikor je konstanta C višja, širša je meja in več napak lahko algoritem tolerira.

V našem delu smo klasificirali večdimenzionalne vektorje z večrazrednim klasifikatorjem, torej nismo uporabljali linearnih klasifikatorjev, temveč nelinearne. Metoda podpornih vektorjev je razširitev klasifikatorja podpornih



Slika 5.22: Slika prikazuje razliko med večjo (leva slika) in manjšo (desna slika) konstanto C. Z večjo konstanto C, kjer bo tudi meja širša, je večja možnost napačne klasifikacije [12].

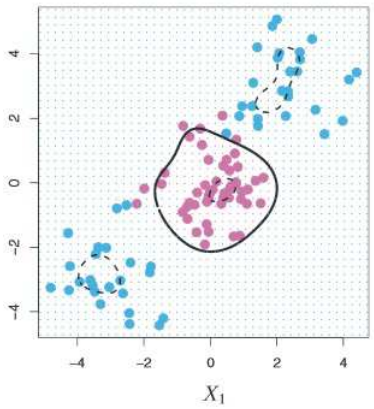
vektorjev, ki z uporabo jedra omogoča nelinearno ločevanje med razredi.

Na uporabljenih podatkovnih zbirkah slik teksta naravnih scen smo evalvirali 3 različna jedra SVM klasifikatorja in sicer linearno jedro, RBF (angl. Radial Basis Function) jedro in Chi-Squared jedro. Linearno jedro (5.17) omogoča hitrejšo klasifikacijo in hitrejše učenje na večjih učnih množicah, saj je računsko manj zahtevno kot bolj kompleksna jedra, vendar smo na račun hitrosti prikrajšani za točnost. Višjo klasifikacijsko točnost smo dosegli z RBF jedrom (5.18), ki je nelinearno jedro.

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j} \tag{5.17}$$

$$K(x_i, x_{i'}) = exp\left(\frac{\|x_i - x_{i'}\|^2}{2\sigma^2}\right) \tag{5.18}$$

Še boljše rezulate pa smo dosegli s Chi-Squared jedrom (5.19), ki je nabolj primerno za klasifacijo histogramov. Chi-Squared jedro predvideva nenegativne podatke, zato moramo predhodno pretvoriti vse negativne vrednosti v pozitivne, obenem pa smo morali podatke tudi normalizirati z L1-norm me-



Slika 5.23: Nelinearno RBF jedro [12].

todo. Normalizacijo podatkov je možno racionalizirati s povezavo Chi-Squared razdalje, ki je razdalja med diskretnimi porazdelitvami verjetnosti.

$$K(x_i, x_{i'}) = 1 - \sum_{i=1}^{n} \frac{2 \cdot (x_i - x_{i'})^2}{(x_i + x_{i'})} \tag{5.19}$$

RBF in Chi-Squared jedra ločujeta podatke nelinearno, kot je demonstrirano na sliki 5.23 in dosegata višjo klasifikacijsko točnost v primerjavi z linearnim jedrom, vendar sta računsko zahtevnejša. V primeru prevelikega števila značilk vektorja linearno jedro dosega višje klasifikacijske rezultate v primerjavi z RBF jedrom.

Razpoznava znakov je večrazredni problem, saj v našem primeru uporabljamo 52 razredov oz. 26 razredov v primeru invariantnosti na velikost znaka. S hiperravnino lahko ločimo le 2 razreda, kjer je naš klasifikator linearen. Med popularnejšimi metodami, kjer lahko SVM algoritem rešuje večrazredne probleme sta "eden proti vsem" ter "eden proti enemu". S pristopom "eden proti enemu" konstruiramo $\binom{K}{2}$ SVM modelov (K število razredov) in jih primerjamo med seboj. S pristopom "eden proti vsem", ki je alternativni pristop reševanja večrazrednih problemov, umestimo (angl. fit) K SVM modelov vsakič, ko primerjamo enega izmed K razredov s preostalimi K-razredi.

Parametri

SVM algoritem je del knjižnice Scikit-learn in omogoča izbiro slednjih parametov "gamma", kar definira, kako daleč sega vpliv enega učnega vzorca, kjer nižje vrednosti pomenijo "daleč" in višje "blizu". Z majhnim "gamma" parametrom je lahko naš model preveč omejen in ne more zajeti pravilne oblike naših podatkov. Nenegativna konstanta C definira, kako visoke napake oz. napačne klasifikacije sprejme naš model. Optimalen C lahko izračunamo s prečnim preverjanjem.

5.3.2 K-NN

K-NN klasifikator je preprosta statistična metoda, ki dobro deluje na večih problemih. Klasifikator K-NN deluje na principu Bayesovega klasifikatorja, vendar v primeru metode K-NN ne poznamo pogojne porazdelitve Y glede na X, torej Bayesova klasifikacije ni mogoča, zato K-NN metoda ocenjuje pogojne porazdelitve Y in dane vrednosti X ter na podlagi tega klasificira vzorec v razred z navišjo ocenjeno verjetnostjo. Glede na podano pozitivno vrednost K in opazovanim vzorcem x, poskuša K-NN klasifikator identificirati K točke v učni množici, ki so najbližje vzorcu x, opisane z η_0. Nato oceni pogojno verjetnost za razred j kot del točk η_0, katerih odziv je enak j [12].

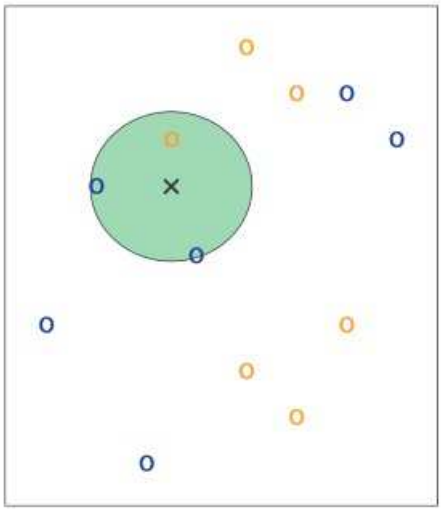
Učna množica metode K-NN predstavlja matriko 1-dimenzionalnih vektorjev značilk na podlagi katere lahko klasificiramo vzorec z računanjem razdalje od posameznega vzorca oz. vrstice v tej matriki. Metoda K-NN zahteva shrambo celotne podatkovne množice, kar lahko vodi do prevelike računske zahtevnosti v primeru, da je učna množica prevelika.

Ker metoda K-NN izračuna razdaljo med opazovanim vzorcem in med učnimi vzorci, lahko tu apliciramo različne algoritme za izračun te razdalje, kot so Evklidska razdalja ter Jaccard, Mahalanobis, Canberra, Manhattan, Minkowski. Najpogosteje se uporablja Evklidska razdalja, vendar se je v našem primeru izkazalo, da Manhattanska razdalja, ki je definirana z enačbo (5.20), prinese najboljše rezultate.

$$d(p,q) = \|p - q\| = \sum_{i=1}^{n} |p_i - q_i| \tag{5.20}$$

Kjer sta (p, q) vektorja:

$$p = (p_1, p_2, \ldots, p_n) \text{ in } (q_1, q_2, \ldots, q_n)$$



Slika 5.24: Vizualizacija metode K-NN, kjer je K enak 3 [12].

Parametri

Eden izmed parametrov, ki ga lahko definiramo v metodi K-NN je način izračunavanja razdalj med opazovanim vzorcem in učnimi vzorci. Pomemben parameter je tudi število sosedov opazovanega vzorca, s katerim lahko povečamo klasifikacijsko točnost. V primeru, da je K enak 1, je število najbližjih sosedov opazovanega vzorca enako 1, v primeru, da je K enak 3 bo K-NN algoritem najprej identificiral vse tri sosede opazovanega vzorca, ki so mu po algoritmu razdalje najbližji. Kot je razvidno na sliki 5.24 so v krogu dva modra vzorca in en rumen vzorec, torej bo črn vzorec (križ) z verjetnostjo 2/3 pripadal razredu modrih krogcev in z verjetnostjo 1/3 bo pripadal rumenim krogcem. Izračunamo lahko torej s kakšno verjetnostjo pripada opazovani vzorec določenemu razredu, v kolikor je K enako številu vseh preučevanih razredov.

5.3.3 Naivni Bayes

Naivni Bayesov klasifikator spada med verjetnostne klasifikatorje in aplicira Bayesov teorem z naivnimi predpostavkami med elementi. Bayesov teorem verjetnosti opisuje verjetnost nekega dogodka na podlagi pogojev, ki so lahko povezani s tem dogodkom in je matematično opisan z naslednjo enačbo (5.21). Naivni Bayes vsebuje predpostavko o neodvisnosti med napovedmi, kar pomeni, da predpostavlja, da prisotnost določene značilnosti nekega razreda ni povezana s prisotnostjo katerekoli druge značilnosti. V našem primeru bi to pomenilo, da je znak črka "O", če je okrogla, črne barve in je znotraj prazna. Četudi so omenjene lastnosti odvisne ena od druge, neodvisno prispevajo k verjetnosti da je znak črka "O", zaradi tega razloga pravimo naivnemu Bayesu "naiven".

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{5.21}$$

Kjer sta A in B dogodka:

$P(c|x)$ posteriorna verjetnost razreda c ob danih napovedi/atributih

$P(c)$ predhodna verjetnost razreda

$P(x|c)$ verjetnost dogodka ob danem razredu

$P(x)$ predhodna verjetnost atributa

Pomankljivost naivnega Bayesa je ta, da v kolikor se pojavi neznan vzorec, ki predhodno ni bil del učne množice (npr. znak Y ni del slovenske abecede) bi mu NB dodelil verjetnost 0, saj ne more izračunati njegove napovedi. V izogib temu, lahko uporabimo metodo glajenja z Laplacovim ocenjevanjem. Prednost naivnega Bayesovega klasifikatorja je ta, da je računsko nezahteven pri obdelavi večrazrednih napovedi, kar pomeni, da je primeren za realnočasovne aplikacije. Dosega boljše rezultate z majhnimi učnimi množicami. Na splošno

pa ne dosega dobrih klasifikacijskih rezultatov v primerjavi z ostalimi klasifikacijskimi metodami. Izbran je bil kot "ground truth" klasifikator, na podlagi katerega smo razlikovali uspešnost ostalih. Naivni Bayes je še vedno relevanten na področju strojnega učenja, saj je v nekaterih primerih primerljiv z metodami podpornih vektorjev, predvsem na področju večrazredne klasifikacije teksta.

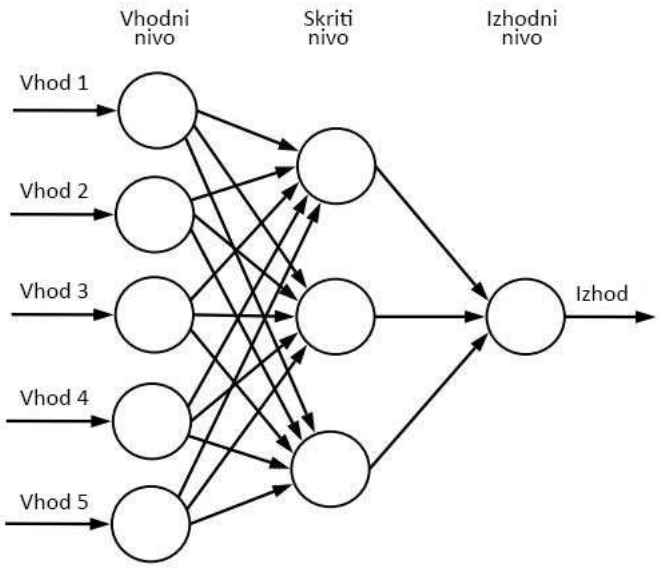
Parametri

Z uporabo knjižnice Scikit-learn lahko zgradimo različne modele naivnega Bayesa. V našem delu smo uporabili gausov NB, multinomski NB in Bernoullijev NB. Gaussov NB model prejme parameter σ (sigma), s katerim je definirana varianca vsake značilnosti razreda. Pri multinomskem in Bernoullijevim NB modelom nismo definirali parametra α (alpha) (stopnja glajenja - Laplace), saj smo v našem primeru kategorizirali poznane vzorce.

5.3.4 ANN

Umetne nevronske mreže (ANN) [28] simulirajo biološki nevrološki sistem, kot so možgani, kjer procesirajo informacije. Ključno pri procesiranju podatkov je to, da je sistem zgrajen iz več med seboj povezanih procesnih elementov (nevronov), ki delujejo usklajeno za reševanje specifičnih problemov. ANN sistemi se kot ljudje učijo na primerih. Nevronske mreže so uporabljene za razpoznavo vzorcev in detekcijo gibanj, ki so za ljudi preveč kompleksni, da bi jih razpoznali. ANN je verjetnostni klasifikator, kar pomeni, da nam poda rezultat v obliki verjetnosti. V primeru klasifikacije posameznega znaka nam ANN pove, s kakšno verjetnostjo ta znak pripada določenemu razredu.

V našem delu smo uporabili implementacijo ANN klasifikatorja, ki temelji na večnivojskem "perceptronu" oz. predkrmiljeni mreži (angl. feed-forward network). Predkrmiljene mreže omogočajo potovanje signalov samo v eno smer, od vhoda do izhoda. Najosnovnejši gradnik umetne nevronske mreže je "perceptron", ki je sestavljen iz enega ali več vhodov ter enega izhoda. Izhod oz. rezultat perceptrona je definiran z aktivacijsko funkcijo. Perceptron



Slika 5.25: Večnivojska umetna nevronska mreža [12].

(matematični model biološkega nevrona) upošteva model predkrmiljene mreže, kar pomeni, da so vhodi, posredovani nevronu, procesirani in nato podani kot rezultat. Vsak vhod perceptrona moramo predhodno utežiti, utež pomnožimo z vhodom in nato oba rezultata seštejemo ter vrnemo rezultat glede na aktivacijsko funkcijo. Perceptron ima poleg vhodnih parametrov še dodaten vhodni pristranski (angl. bias) parameter, ki je vedno prisoten in tudi vsebuje utež. V našem primeru so vhode predstavljali vektorji značilk.

Večnivojski perceptron je sestavljen iz večih nivojev nevronov, kot prikazuje slika 5.25, ki predstavljajo usmerjen graf, kjer je vsak nivo povezan z naslednjim in navadno sestavljen iz treh nivojev. Na prvem nivoju so nevroni, ki prejmejo vhodne podatke in izdelujejo preproste odločitve, na podlagi uteževanja vhodnih podatkov. Na drugem oz. skritem (niso del vhoda niti izhoda) nivoju so nevroni, ki ravno tako izvajajo odločitve, vendar z rezultati prvega nivoja, s čimer ustvarjajo bolj kompleksne in sofisticirane odločitve. Še bolj kompleksne odločitve pa izvaja izhodni oz. zadnji nivo. Prvi nivo prejme

informacije, ki so posredovane naprej celotnemu omrežju. Delovanje skritih nevronov je določeno z dejavnostjo vhodnih nevronov in uteži na povezavah med vhodnimi in skritimi nevroni. Delovanje izhodnega nivoja je določeno z dejavnostjo skritih nevronov in uteži med skritimi in izhodnimi nevroni. Prednost večnivojskega perceptrona je ta, da je sposoben reševati nelinearne probleme. Za učenje klasifikatorja smo uporabili povratni algoritem (angl. backpropagation), ki optimizira uteževanje posameznega nevrona in s tem posreduje podatke, koliko posamezen nevron prispeva k splošni napaki nevronske mreže. Pri učenju se torej generira vrednost napake, ki je poslana nazaj skozi omrežje in ponovno popravi uteži vseh povezav, pri tem lahko algoritem avtomatično optimizira uteži.

Večnivojske nevronske mreže uporabljajo sigmoidne ali tanh (5.23) aktivacijske funkcije, ki smo jih tudi mi uporabili v naši implementaciji, saj rešujejo nelinearne probleme in lahko vsebuje več skritih nivojev. Sigmoidna unkcija je definirana s formulo opisano v enačbi (5.22). Sigmoidni nevroni prav tako kot perceptroni vsebujejo uteži, vendar z njimi veliko lažje ugotovimo, kako bodo uteži in pristranske vrednosti (angl. bias) spremenile izhod.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5.22}$$

$$tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{5.23}$$

Parametri

Uporabili smo algoritem ANN, ki je del knjižnice Scikit-learn. Klasifikator je implementiran tako, da mu lahko spreminjamo različne parametre. Eden izmed parametrov je število skritih nivojev, ki definira število nivojev med vhodnim in izhodnim nivojem ter število nevronov na posameznem nivoju. Število skritih nivojev lahko določimo tako, da sklenemo kompromis med številom skritih nivojev in časom, ki ga potrebujemo za učenje nevronskega modela. V kolikor

želimo zmanjšati čas učenja, lahko zmanjšamo število skritih nivojev in epoh, vendar se z večjim številom nevronov ponavadi pridobi višje klasifikacijske rezultate. Število epoh definira število iteracij učenja našega ANN modela, kjer izračunamo nove uteži in pristranske vrednosti (angl. bias), v našem primeru smo definirali 100 epoh. Zaradi dodeljevanja naključnih uteži posameznim povezavam na začetku, algoritem ANN ne generira vedno istega rezultata.

Poglavje 6

Rezultati

Na podatkovnih zbirkah slik teksta ICDAR 2003 [4], Chars74K [6], CVL OCR DB [5] ter sintetično generirani zbirki slik teksta, smo evalvirali algoritme HOG, PHOG in Co-HOG. Omenjene algoritme smo implementirali na podlagi virov Navneet Dalal et. al. [17], Z. R. Tan et. al [36] ter S. Tian [37]. Klasifikacijske algoritme smo uporabili iz knjižnic opisanih v poglavju 3.1.3.

6.1 Priprava podatkov

Podatkovni zbirki slik teksta ICDAR 2003 [4] in Chars74K [6] vsebujeta znake, ki predstavljajo števila. Slednje smo iz podatkovnih zbirk izločili, s čimer se je naš klasifikacijski problem zmanjšal na 52 razredov. Podatkovna zbirka ICDAR 2003 [4] vsebuje ločila, zbirka CVL OCR DB [5] pa nekaj šumnikov. Slike znakov, ki so predstavljale ločila ali šumnike, smo tako kot številke odstranili iz klasifikacijskega problema, saj smo se omejili na znake angleške abecede.

Slike, vsebovane v podatkovnih zbirkah slik teksta, so različnih velikosti. Zaradi potrebe po pridobitvi vektorjev značilk uniformnih velikosti, smo morali normalizirati slike znakov podatkovnih zbirk. Slike smo normalizirali na velikost 32×32 s filtrom za glajenje krivulj (angl. anti-aliasing), s čimer smo

Slika 6.1: Normalizacija znaka "E". Slika a) prikazuje originalno sliko, slika b) pa normalizirano sliko.

sliko dodatno zgladili. Z normalizacijo znakov smo lahko pridobljene vektorje značilk klasificirali brez algoritmov za redukcijo značilk, saj so bili vektorji že enakih dolžin. Pri normalizaciji smo v nekaterih primerih izgubili določene informacije, kjer je prihajalo do distorzije in deformacije slike znaka, kot je razvidno na sliki 6.1. Deformacija slike znaka je doprinesla k zmanjšanju klasifikacijske točnosti. Z normalizacijo smo tako nekatere slike proporcionalno povečali oz. zmanjšali, preostale pa deformirali.

6.2 Pridobivanje značilk

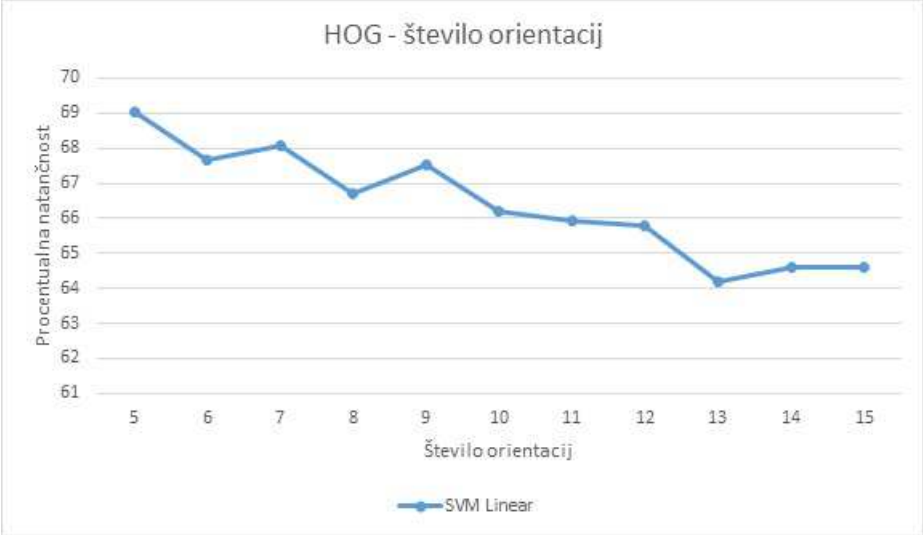
Implementirali smo tri različne algoritme za pridobivanje značilk iz slik znakov in sicer, HOG, PHOG ter Co-HOG. Vsi trije algoritmi temeljijo na metodah pridobivanja značilk brez predprocesiranja in segmentacije.

Pri izračunavanju gradientov smo primerjali različne konvolucijske matrike in način obravnavanja robnih primerov. Za izračun gradientov smo uporabili preprosto 1-dimenzionalnost matriko 5.4 in Sobelov operator 5.5. Izkazalo se je, da je preprosta konvolucijska matrika doprinesla k večji klasifikacijski točnosti, saj se je povprečna klasifikacijska točnost razlikovala od o.5 % do 1.0 % v prid preproste konvolucijske matrike. Test smo opravili na algoritmih

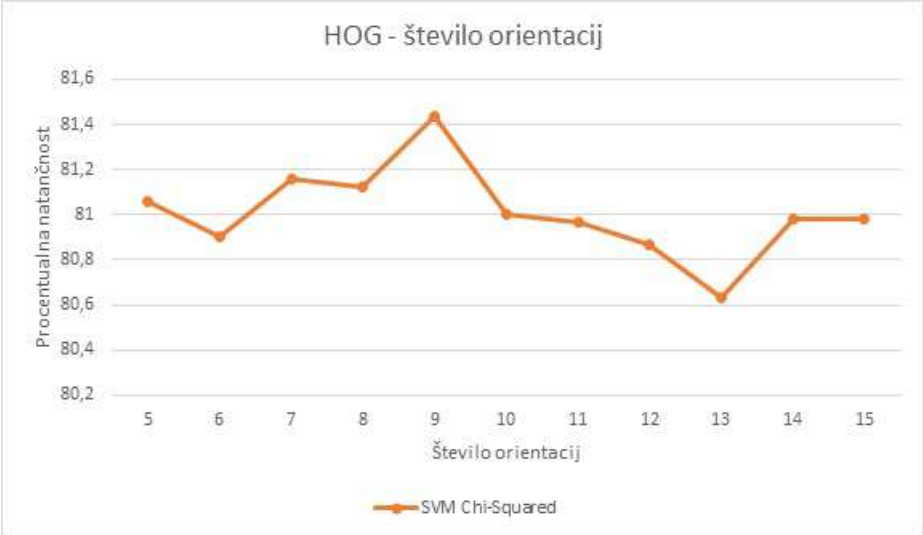
HOG, PHOG in Co-HOG ter ICDAR 2003 podatkovni zbirki slik. Klasifikacijsko točnost smo izračunali s klasifikatorjem SVM v povezavi s Chi-Squared jedrom. Evalvirali smo tudi različne načine obravnavanja robnih primerov pri konvoluciji, kjer je način "najbližji" (angl. nearest mode) prinesel najboljše klasifikacijske rezultate. Na podlagi te predpostavke smo ta način obravnavanja robnih primerov uporabljali tudi pri ostalih dveh algoritmih, torej PHOG in Co-HOG. Način obravnavanja robnih primerov je v nekaterih primerih ogromno doprinesel k izboljšanju rezultatov, saj se je klasifikacijska točnost med načinom "najbižji" in načinom "konstanta" (angl. constant) razlikovala za 5 %. Do tako velikih razlik je prihajalo zaradi majhne velikosti slik znakov, ki so bile normalizirane na 32×32 slikovnih elementov.

Za določitev optimalnega števila orientacij pri metodi HOG smo evalvirali algoritem na podatkovni zbirki ICDAR s klasifikatorjem SVM, kjer smo uporabili dve različni jedri in sicer linearno ter Chi-Squared. SVM z linearnim jedrom je, kot je razvidno na grafu slike 6.2, dosegal nižjo klasifikacijsko točnost, zato smo uporabilit meritve s Chi-Squared jedrom 6.3, kjer je razvidno, da je optimalno število orientacij gradienta 9. Razlike med orientacijami so minimalne, od 0.2 % do 0.8 %. Razpon stopinj, ki jih je histogram obravnaval je segal od 0° do 180°. Po optimizaciji števila orientacij smo analizirali, kako velikosti posameznih celic vplivajo na klasifikacijsko točnost algoritma. Algoritem HOG smo testirali pri različnih velikosti celic, kot je razvidno na grafu slike 6.4, kjer smo uporabili 9 orientacij in 3×3 velikosti blokov. Iz grafa slike 6.4 je razvidno, da nam najvišjo klasifikacijsko točnost prinesejo celice velikosti 8×8. Zadnji parameter metode HOG je velikost posameznega bloka, kjer velikost bloka definira število celic v bloku. Na grafu slike 6.5 je razvidno, da velikosti blokov 4×4 celic prinesejo najvišjo klasifikacijsko točnost, kar pri velikosti slike znakov 32×32 slikovnih elementov in velikosti celic 8×8 slikovnih elementov pomeni, da je dolžina vektorja značilk skoraj za polovico manjša, kot pri blokih velikosti 3×3, vendar vsebuje bolj relevantne podatke.

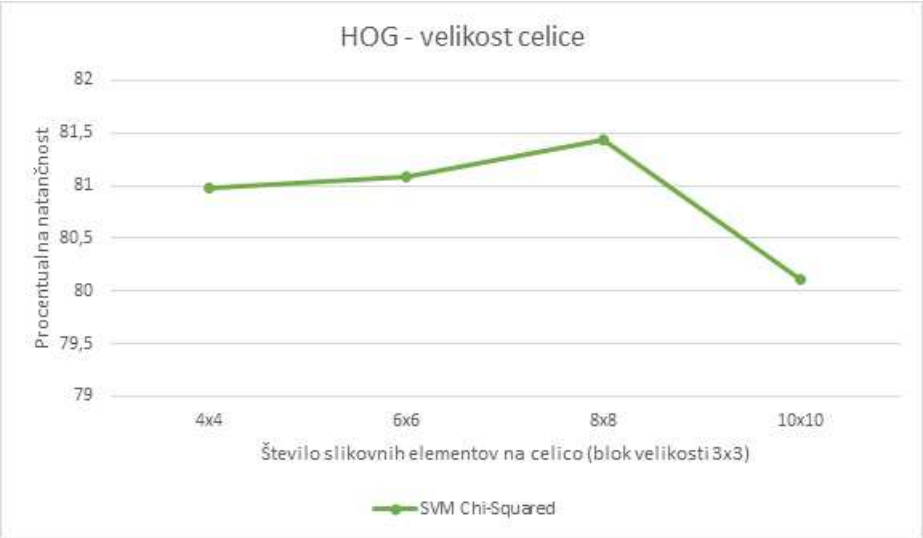
Pri algoritmu PHOG smo število orientacij povzeli iz algoritma HOG in



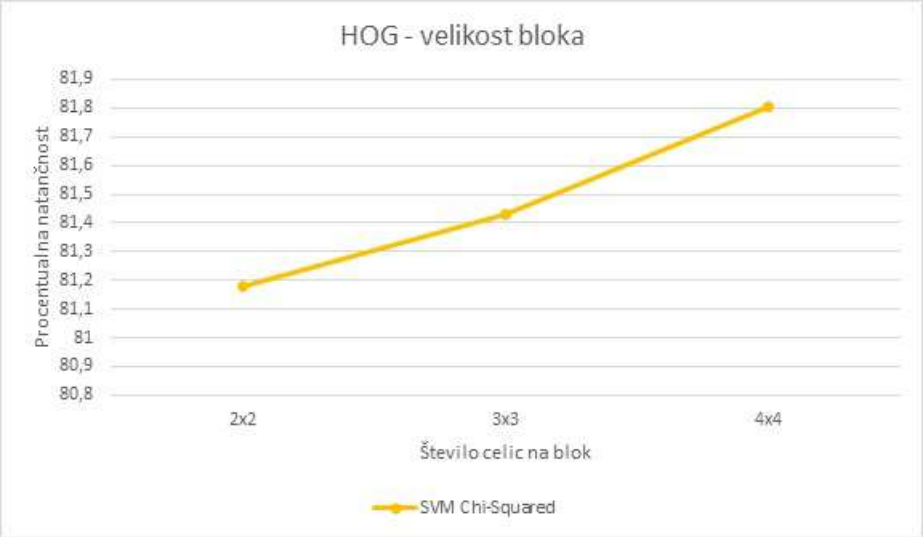
Slika 6.2: Klasifikacijska točnost algoritma HOG na zbirki slik ICDAR pri različnih orientacijah. Za klasifikacijo je bil uporabljen SVM z linearnim jedrom.



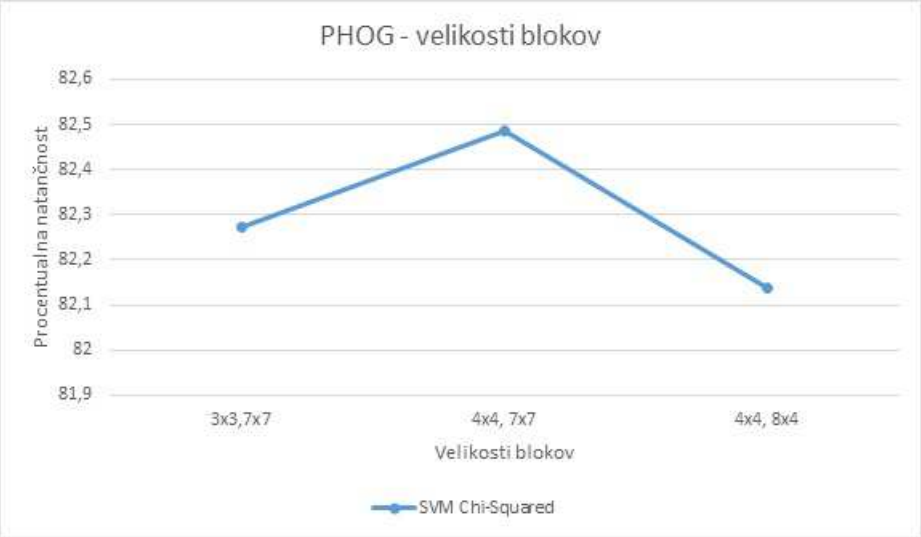
Slika 6.3: Klasifikacijska točnost algoritma HOG na zbirki slik ICDAR pri različnih orientacijah. Za klasifikacijo je bil uporabljen SVM s Chi-Squared jedrom.



Slika 6.4: Klasifikacijska točnost algoritma HOG na zbirki slik ICDAR pri različnih velikostih celic.



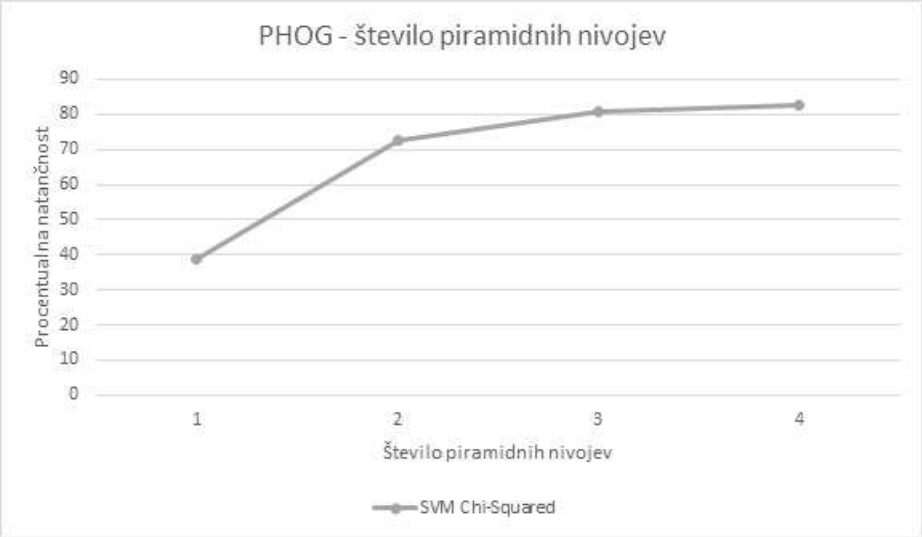
Slika 6.5: Klasifikacijska točnost algoritma HOG na zbirki slik ICDAR pri različnih velikostih blokov.



Slika 6.6: Klasifikacijska točnost algoritma HOG na zbirki slik ICDAR ob različnih velikosti blokov, kjer sta na prvem in drugem nivoju velikost blokov definirani z 1×1 in 2×2, tretji in četri nivo sta prikazana na x osi grafa.

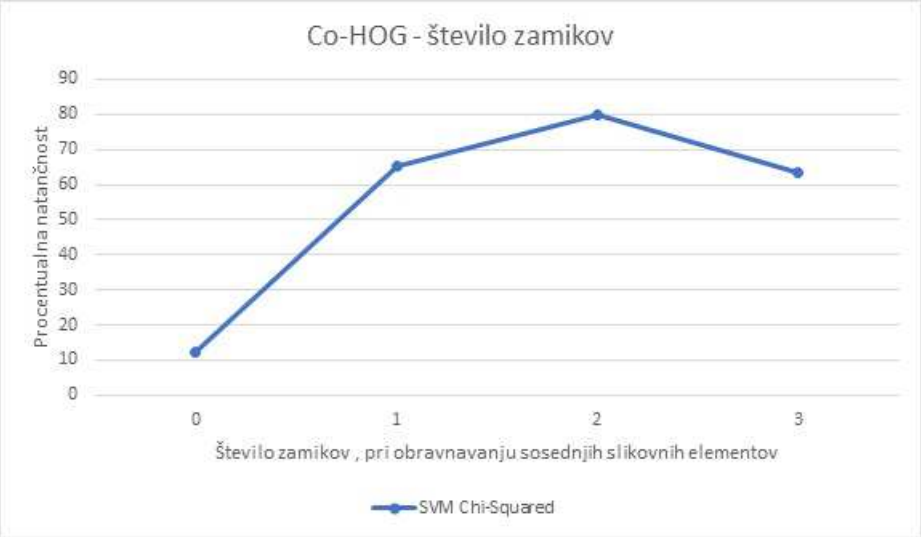
predpostavljali, da je optimalno število orientacij 9, saj je pri algoritmu HOG prinašalo najvišje klasifikacijske rezultate. Velikosti blokov na posameznem piramidnem nivoju smo zaradi analize HOG algoritma, ki je razvidna na grafu slike 6.5 prilagodili tako, da smo na tretjem nivoju piramide modificirali velikost bloka na 4×4, ostali nivoji niso bili spremenjeni in so ostali enaki kot v shemi 5.2.2. Sprememba velikosti blokov tretjega nivoja je povišala klasifikacijsko točnost za 0.3 %. Velikosti celic ni bilo potrebno optimizirati, saj so bile definirane dinamično s formulo (5.9). Število piramidnih nivojev je vplivalo na klasifikacijske rezultate, saj se je z večim številom piramidnih nivojev povečevalo število informacij, ki smo jih pridobili iz slike znaka, kot je razvidno na grafu slike 6.7.

Za algoritem Co-HOG smo prav tako kot v PHOG in HOG 6.2 algoritmih uporabili 9 orientacij za gručenje orientacij gradientov v histogramu. Za število blokov v celotni sliki smo uporabili velikost 4×4. Co-HOG algoritem ne vsebuje prekrivajočih se blokov, zato ni potrebe po definiranju celic, saj te vse-



Slika 6.7: Klasifikacijska točnost algoritma HOG na zbirki slik ICDAR pri različno definiranih piramidnih nivojih.

bujejo samo bloke. Na grafu slike 6.8 je predstavljena klasifikacijska točnost ob različnih zamikih izračunavanja so-pojavitvene matrike, kjer velikost zamika pomeni število bližnjih slikovnih elementov, ki jih referenčni slikovni element smatra za soseda 5.17. Pri izračunavanju gradienta smo uporabili preprosto 1-dimenzionalno matriko in ne Sobelov operator, kot je predlagano v članku [37], saj nam je ta prinesla višje klasifikacijske rezultate. Definirali smo tudi optimalen prag, ki je definiral, katere sosednje pare vrednosti magnitud gradienta lahko obravnavamo v so-pojavitveni matriki, saj so ničelne vrednosti magnitud gradienta doprinesle k neenakomerni uteženosti. Sprejemljive vrednosti magnitud gradienta so bile vrednosti od 0.5 in več.



Slika 6.8: Klasifikacijska točnost algoritma Co-HOG na zbirki slik ICDAR pri različno definiranih zamikih referenčnega slikovnega elementa.

6.3 Klasifikacija

Klasifikacijski algoritmi

Za lažjo in hitrejšo evalvacijo smo ekstrahirane značilke in njihove oznake shranjevali v podatkovne datoteke, ki so bile poimenovane po imenu algoritma za pridobivanje značilk ter imenu podatkovne zbirke iz katere smo pridobili slike znakov. Ob naslednji evalvaciji podatkovne zbirke z istim algoritmom za pridobivanje značilk nam torej ni bilo potrebno ponovno čez postopek pridobivanja značilk, v kolikor smo klasificirali podatke z drugim klasifikacijskim algoritmom. Klasifikacijo smo opravili na 52. razredih pri variantnosti velikosti znaka, oz. na 26. razredih pri invariantnosti velikosti znaka.

Klasifikator SVM smo testirali z linearnim jedrom, ki je hitrejši klasifikator, vendar je ta v povprečju prinašal 28 % nižje klasifikacijske rezultate. Z uporabo metode "eden proti vsem" se je klasifikacijska točnost povišala, a je vseeno v povprečju prinašala 10 % nižje klasifikacijske rezultate. Najboljše rezultate smo pridobili s Chi-Squared jedrom in metodo "eden proti enemu", ki je v

povprečju dosegal za 22 % višjo klasifikacijsko točnost kot SVM z jedrom RBF in enako metodo.

Pri klasifikaciji z algoritmom K-najbližjih sosedov, smo v povprečju dosegali 5 % nižjo klasifikacijsko točnost kot s klasifikatorjem SVM in Chi-Squared jedrom, kar je za tako preprost klasifikator vseeno visoka točnost, saj klasificira hitreje in je primeren za realnočasovne aplikacije. Pri klasifikaciji smo uporabili različne algoritme za izračun razdalje med vzorci. Najvišje klasifikacijske rezultate je dosegal Manhattan algoritem z izračunom razdalje, ostali algoritmi za izračun razdalj, kot je Evklidska, Minkowski pa so v povprečju dosegali 7 % nižjo klasifikacijsko točnost. Pomanjkljivost K-NN klasifikatorja, je ta, da shrani podatke v pomnilnik. Za PHOG in Co-HOG algoritma je K-NN klasifikator problematičen, saj ti dve metodi zahtevata veliko več pomnilniškega prostora kot HOG algoritem, saj so vektorji značilk od 10 do 20-krat večji.

S klasifikatorjem ANN smo dosegali nižje klasifikacijske rezultate kot s klasifikatorjema SVM, kjer smo v povprečju dosegali 4 % nižje klasifikacijske rezultate, kot pri klasifikatorju SVM s Chi-Squared jedrom. Pri klasifikatorju ANN smo z večjim številom skritih nivojev in iteracij dosegali boljše rezultate, vendar za ceno računske zahtevnosti, saj je večje število skritih nivojev in večje število iteracij zahtevalo daljše procesiranje podatkov. Na podlagi tega smo določili optimalno mero, kjer je smo definirali 80 skritih nivojev ter 150 iteracij oz. epoh. Kot aktivacijsko funkcijo smo uporabili tanh (5.23) funkcijo ter optimizacijski algoritem, ki temelji na stohastičnem gradientu. Zaradi naključne začetne vrednosti za določanje uteži, smo pri vsaki klasifikaciji dobili malenkost različne rezultate, ki so se razlikovali za približno 0.2 %.

Klasifikacijska metoda Naivni Bayes nam je povprečno prinašala najslabše klasifikacijske rezultate in sicer 20 % nižjo klasifikacijsko točnost pri vseh algoritmih za pridobivanje značilk, vendar je bila najhitrejša klasifikacijska metoda. Najboljše rezultate nam je prinašal Gaussov model porazdelitve, ki gruči podatke enakih razredov.

Podatkovne zbirke slik teksta

Klasifikacijske algoritme smo evalvirali na podatkovnih zbirkah slik naravnih scen ICDAR, Chars74K, CVL OCR DB ter sintetični zbirki. Vsako podatkovno zbirko smo klasificirali z različnimi klasifikatorji, in sicer s klasifikatorjem SVM, K-NN, ANN ter NB. Tabele od 6.1 do 6.5 prikazujejo klasifikacijsko točnost v povezavi z različnimi algoritmi pridobivanja značilk in klasifikatorji. V tabelah so z zeleno označeni dobri rezultati ter z rdečo slabi. Klasifikacijske rezultate vsakega algoritma smo razdelili na dva dela, torej na zgornjo in spodnjo vrstico, kjer zgornja vrstica predstavlja klasifikacijsko točnost pri uporabi invariantnosti glede na velikost znakov, kjer klasifikacijski problem vsebuje 26 razredov. Spodnja vrstica pa prikazuje klasifikacijsko točnost 52. razredov, kar pomeni da klasificiramo tako male kot velike znake.

Najprej smo evalvirali podatkovno zbirko slik Chars74K, kot prikazuje tabela 6.1, ki vsebuje množico slik s slikami znakov slabše kvalitete in množico slik s slikami znakov dobre kvalitete. Množica kvalitetnih slik vsebuje 7700 slik, množica slik slabših kvalitet pa 5000 slik znakov. Podatkovno zbirko smo evalvirali, tako da smo za učno množico vzeli slike znakov dobre kvalitete, za testno množico pa slike znakov slabše kvalitete. Najvišje klasifikacijske rezultate na podatkovni zbirki Chars74K nam je prinesel algoritem PHOG v povezavi s klasifikatorjem SVM z Chi-Squared jedrom pri invariantnosti velikosti znaka, in sicer 70,02 %, pri razlikovanju velikosti znaka pa 63,91 %. Pri tem smo parameter C, ki pri SVM klasifikatorju pomeni toleranco napake, nastavili na 1, saj so nam manjše vrednosti prinesle slabše klasifikacijske rezultate. Najslabše rezultate nam je prinesel algoritem HOG v povezavi s klasifikatorjem NB in sicer 50,53 % pri invariantnosti velikosti znaka ter 43,36 % pri razlikovanju velikosti znakov, kjer je klasifikacijski problem razdeljen na 52 razredov. V povprečju je najslabšo klasifikacijsko točnost na zbirki slik Chars74K dosegal algoritem Co-HOG.

Evalvacija podatkovne zbirke CVL OCR DB je bila opravljena s prečnim preverjanjem K-fold [12], kot je razvidno na tabeli 6.2, kjer je bil K enak

10. Pri navzkrižnem preverjanju smo razdelili celotno podatkovno zbirko na 10 enako velikih delov in za učno množico vzeli 9 delov oz. $K - 1$, za testno množico pa 1 del oz. 10 % celotne zbirke. Ta postopek smo ponovili 10-krat, tako da smo dobili 10 klasifikacijskih rezultatov, ki smo jih povprečili, da smo dobili realno klasifikacijsko točnost. Zbirka CVL OCR DB vsebuje 7000 slik naravnih scen, kar pomeni, da je velikost učne množice štela 5000 učnih vzorcev, testna množica pa prbližno 500 vzorcev ob vsakem navzkrižnem preverjanju. Najboljše klasifikacijske rezultate je dosegel algoritem PHOG s klasifikatorjem SVM ter Chi-Squared jedrom in sicer 93,23 % klasifikacijsko točnost (84,57 % klasifikacijska točnost pri razlikovanju velikosti znakov). Najslabše rezultate je dosegel algoritem Co-HOG in sicer 81.98 % klasifikacijsko natančnost z NB klasifikatorjem (70,38 % klasifikacijska točnost pri razlikovanju velikosti znakov). Podatkovno zbirko CVL OCR DB smo evalvirali tudi z učno množico ICDAR, prikazano v tabeli 6.3. Povprečna klasifikacijska točnost v tem primeru je bila manjša kot pri prečnem preverjanju CVL OCR DB baze, saj je bila učna množica sestavljena iz popolnoma druge podatkovne zbirke slik teksta. Največjo klasifikacijsko točnost, kjer smo uporabili učne vzorce slik teksta ICDAR in jih testirali na zbirki slik teksta CVL OCR DB, je prinašala metoda PHOG in sicer 85,85 % klasifikacijsko točnost (74,24 % klasifikacijska točnost pri razlikovanju velikosti znakov) v povezavi s klasifikatorjem SVM in Chi-Squared jedrom. Najslabše klasifikacijske rezultate pa je dosegla metoda CO-HOG v povezavi s klasifikatorjem NB in sicer 70,33 % klasifikacijsko točnost (60,84 % klasifikacijska točnost pri razlikovanju velikosti znakov).

Podatkovno zbirko slik teksta ICDAR smo evalvirali s pomočjo učne množice podatkovne zbirke CVL OCR DB in jo testirali na testni množici slik zbirke ICDAR, kot prikazuje tabela klasifikacijskih rezultatov 6.4. Pri tej evalvaciji je najvišje klasifikacijske rezultate dosegal algoritem PHOG v povezavi z K-NN klasifikatorjem in sicer 71,23 % klasifikacijsko točnost pri invariantnosti velikosti znaka (64,88 % klasifikacijska točnost pri razlikovanju velikosti znakov), najnižje klasifikacijske rezultate pa je zopet dosegel algoritem Co-HOG z NB

Chars74K - slike naravnih scen (dobri in slabi primeri)				
	SVM-chi^2	K-NN	ANN	NB
HOG	68,82 % 62,43 %	64,49 % 57,32 %	64,55 % 57,90 %	50,53 % 43,36 %
PHOG	70,02 % 63,91 %	65,80 % 57,85 %	66,24 % 59,32 %	53,81 % 46,89 %
Co-HOG	67,29 % 61,58 %	62,77 % 55,10 %	63,02 % 56,55 %	50,75 % 43,78 %

Tabela 6.1: Tabela rezultatov zbirke slik Chars74K, kjer je učna množica sestavljena iz slik dobrih kvalitet, testna množica pa iz slik slabih kvalitet.

klasifikatorjem, kjer je klasifikacijska točnost pri invariantnost velikosti znaka merila 59,76 % ter 50,93 % pri razlikovanju velikosti znakov. Zbirka ICDAR je sestavljena iz dveh delov in sicer iz množice slik, ki predstavlja učno množico ter množice slik teksta, ki predstavlja testno množico slik. Na podlagi teh podatkov smo lahko evalvirali podatkovno zbirko ICDAR brez prečnega preverjanja. Pri tej evalvaciji je najvišje klasifikacijske rezultate dosegal algoritem PHOG s klasifikatorejm SVM in Chi-Squared jedrom, kjer je bila klasifikacijska točnost 82,49 % pri 26. razredni klasifikaciji ter 78,58 % pri 52. razredni klasifikaciji. Najnižjo klasifikacijsko točnost je dosegala metoda Co-HOG v povezavi s klasifikatorjem NB in sicer 70,56 % točnost pri invariantnosti velikosti znaka in 66,92 % točnost pri razlikovanju velikosti znaka. Tabela 6.5 prikazuje klasifikacijske rezultate evalvacije podatkovne zbirke ICDAR.

CVL OCR DB				
	SVM-chi^2	K-NN	ANN	NB
HOG	91,51 %	89,01 %	90,20 %	83,27 %
	82,54 %	78,42 %	81,36 %	72,32 %
PHOG	93,23 %	91,06 %	91,89 %	85,90 %
	84,57 %	80,72 %	83,01 %	74,66 %
Co-HOG	90,11 %	87,76 %	88,35 %	81,98 %
	81,26 %	76,43 %	80,03 %	70,38 %

Tabela 6.2: Tabela rezultatov zbirke slik CVL OCR DB s prečnim preverjanjem K-fold [12], kjer je K enak 10.

ICDAR - CVL OCR DB				
	SVM-chi^2	K-NN	ANN	NB
HOG	84,33 %	81,56 %	81,95 %	71,55 %
	73,69 %	70,64 %	71,57 %	61,22 %
PHOG	85,85 %	83,22 %	83,67 %	73,20 %
	74,24 %	72,02 %	73,17 %	63,78 %
Co-HOG	83,97 %	80,14 %	80,20 %	70,33 %
	72,74 %	69,94 %	70,33 %	60,84 %

Tabela 6.3: Tabela rezultatov, kjer so bili uporabljeni učni vzorci podatkovne zbirke ICDAR in testirani na podatkovni zbirki CVL OCR DB.

CVL OCR DB - ICDAR				
	SVM-chi^2	K-NN	ANN	NB
HOG	68,91 %	69,77 %	67,57 %	60,36 %
	61,57 %	60,23 %	60,50 %	51,72 %
PHOG	70,11 %	71,23 %	68,55 %	62,45 %
	63,94 %	64,88 %	61,67 %	53,28 %
Co-HOG	67,39 %	68,52 %	66,81 %	59,76 %
	60,87 %	59,14 %	58,90 %	50,93 %

Tabela 6.4: Tabela rezultatov, kjer so bili uporabljeni učni vzorci podatkovne zbirke CVL OCR DB in testirani na podatkovni zbirki ICDAR.

ICDAR (učna množica) – ICDAR (testna množica)				
	SVM-chi^2	K-NN	ANN	NB
HOG	81,80 %	78,07 %	80,65 %	73,42 %
	77,86 %	72,82 %	76,53 %	68,80 %
PHOG	82,49 %	78,27 %	81,06 %	75,99 %
	78,58 %	73,23 %	76,61 %	71,30 %
Co-HOG	80,33 %	77,26 %	79,85 %	70,56 %
	76,85 %	70,96 %	75,66 %	66,92 %

Tabela 6.5: Tabela rezultatov, kjer so bili uporabljeni učni vzorci podatkovne zbirke ICDAR in testirani na podatkovni testni množici podatkovne zbirke ICDAR.

6.4 Ugotovitve

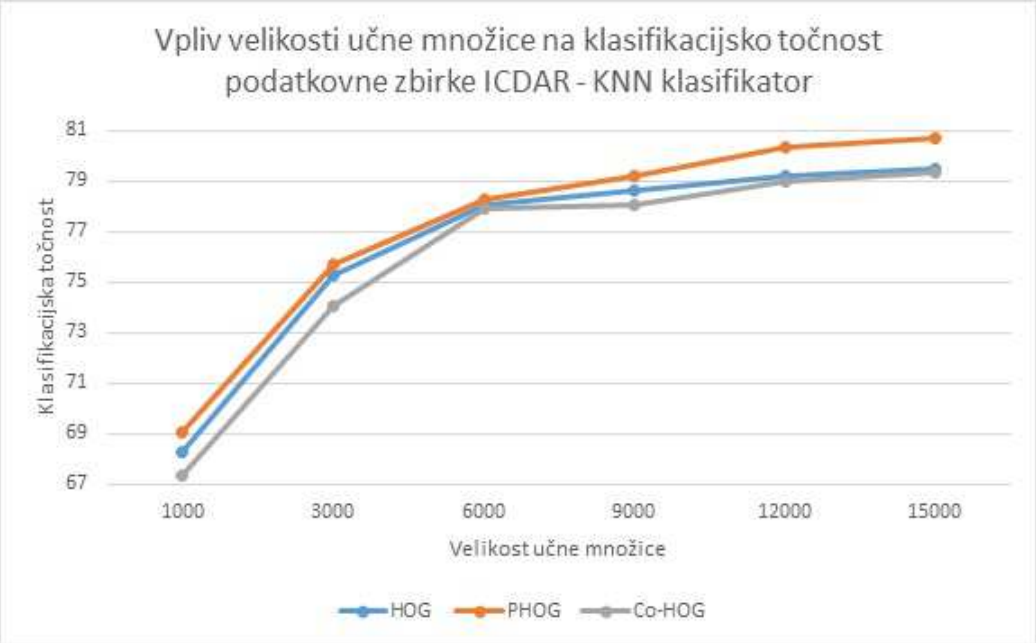
Naš klasifikacijski problem smo omejili na 26 in 52 razredov, kjer pri prvem nismo razlikovali med velikimi in malimi znaki, smo pa upoštevali velikost znaka. V povprečju je velikost znaka vplivala na klasifikacijsko točnost, kjer je bil klasifikacijski rezultat v povprečju od 4 % do 11 % nižji v primeru razlikovanja velikosti znaka. Nihanja med klasifikacijskimi natančnostmi lahko pripisujemo različnim kvalitetam slik znakov posamezne podatkovne zbirke. Na podatkovni zbirki ICDAR, smo z algoritmom PHOG in klasifikatorjem SVM (Chi-Squared jedro) dosegli 82,49 % klasifikacijsko točnost pri invariatnosti velikosti znaka. V metodi, ki je opisana v članku "Using pyramid of histogram of oriented gradients on natural scene text recognition" [36] in na podlagi katerega smo implementirali algoritem PHOG je bila dosežena 82,7 % klasifikacijska točnost pri invariantnosti velikosti znaka ter vključevanju številk. Vendar je učna množica štela 24000 učnih vzorcev. V našem primeru je bila učna množica skoraj 4-krat manjša in šteje 6000 učnih vzorcev. Nižje klasifikacijske rezultate nam je prinesel algoritem Co-HOG, ki smo ga implementirali na podlagi članka "Scene Text Recognition Using Co-occurrence of Histogram of Oriented Gradients" [37], kjer smo na podatkovni zbirki ICDAR v povezavi s klasifikatorjem SVM (Chi-Squared jedro) dosegli 80,33 % klasifikacijsko točnost. Pri invariantnosti velikosti znaka, v omenjenem delu pa je ta 83,6 % prav tako pri invariatnosti velikosti znaka. Vendar je pri slednjem uporabljena podatkovna zbirka velikosti 18500 vzorcev, v našem delu pa učni del zbirke ICDAR šteje 6000 vzorcev, iz česar lahko sklepamo, da bi se potencialno lahko približali željenemu klasifikacijskem rezultatu.

Najvišji klasifikacijski rezultati so bili doseženi na podatkovni zbirki slik teksta CVL OCR DB, kjer smo uporabili prečno preverjanje. Na zbirki CVL OCR DB smo tako dosegli najvišjo klasifikacijsko točnost z algoritmom PHOG in klasifikatorjem SVM (Chi-Squared jedro) in sicer 93,23 % pri invariantnosti velikosti znaka ter 84,57 % pri razlikovanju velikosti znakov. Dobre rezul-

tate lahko pripisujemo kvalitetnim slikam, ki jih je vsebovala omenjena zbirka. Najnižje rezultate z algoritmom PHOG in klasifikatorjem SVM (Chi-Squared jedro) smo dobili z evalvacijo podatkovne zbirke Chars74 in sicer 70,02 % pri invariatnosti velikosti znaka ter 63,91 % pri razlikovanju velikosti znakov. Podatkovna zbirka Chars74K vsebuje slike zelo slabih kvalitet v primerjavi z ostalimi podatkovnimi zbirkami slik teksta.

Podatkovno zbirko ICDAR smo evalvirali z različnimi velikostmi učnih množic, kjer smo želeli prikazati korelacijo med velikostjo učne množice in klasifikacijsko točnostjo, kot je razvidno na grafu slike 6.9. Standardne podatkovne zbirke slik ne vsebujejo željenih velikosti podatkov, zato smo konkatenirali podatke večih učnih množic in jih združili v eno. Z združevanjem večih učnih množic smo tako lahko generirali učno množico, ki je štela 15000 učnih vzorcev in je bila skupek podatkovne zbirke ICDAR (6000 slik znakov - učnega dela podatkovne zbirke), Chars74K (7000 slik znakov dobrih primerov) ter zbirke CVL OCR DB (7000 slik znakov), ki v povprečju vsebuje boljše primere slik znakov kot drugi dve zbirki). Za klasifikacijo podatkov smo izbrali klasifikator K-NN, saj je dosegal dobre klasifikacijske rezultate in zahteval nizko časovno zahtevnost. Iz grafa slike 6.9 lahko razberemo, da je naboljšo klasifikacijsko točnost dosegal algoritem PHOG pri 15000 učnih vzorcih. Na grafu je razvidno, da se je klasifikacijska točnost najvišje povzpela, ko je učna množica vsebovala 3000 učnih vzorcev. Krivulja grafa je nato počasi stagnirala in povzela obliko logaritemske krivulje. Optimalna velikost učne množice glede na klasifikacijsko točnost in časovno zahtevnost procesiranja podatkov, naj bi štela od 9000 do 12000 slik znakov, saj se rezultati nad to vrednostjo ne zvišujejo več drastično.

Prav tako smo ugotovili, da prečno preverjanje v povprečju prinese boljše klasifikacijske rezultate, kar prikazujeta tudi tabeli 6.2 in 6.3, kjer je klasifikacijska točnost višja v povprečju za 8 % do 9 %. Višjo klasifikacijsko točnost lahko pripisujemo kvaliteti slik znakov v zbirki slik znakov, ki je v tem primeru zbirka CVL OCR DB vsebovala bolj kakovostne slike teksta kot podatkovna



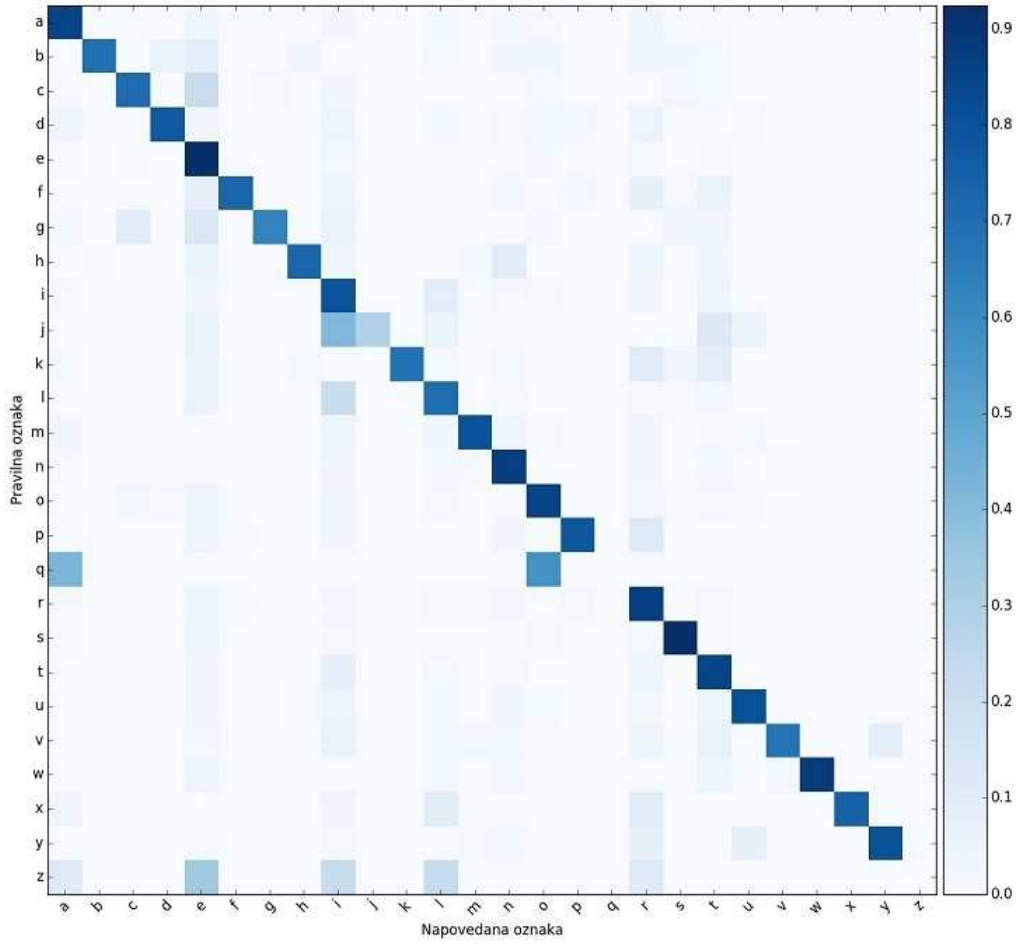
Slika 6.9: Graf slike prikazuje korelacijo med velikostjo učne množice in klasifikacijsko točnostjo. Učna množica je bila skupek večih podatkovnih zbirk in sicer zbirke ICDAR, Chars74K (dobri primeri slik) ter podatkovne zbirke CVL OCR DB. Klasifikacija je bila opravljena s K-NN klasifikatorjem.

zbirka slik znakov ICDAR.

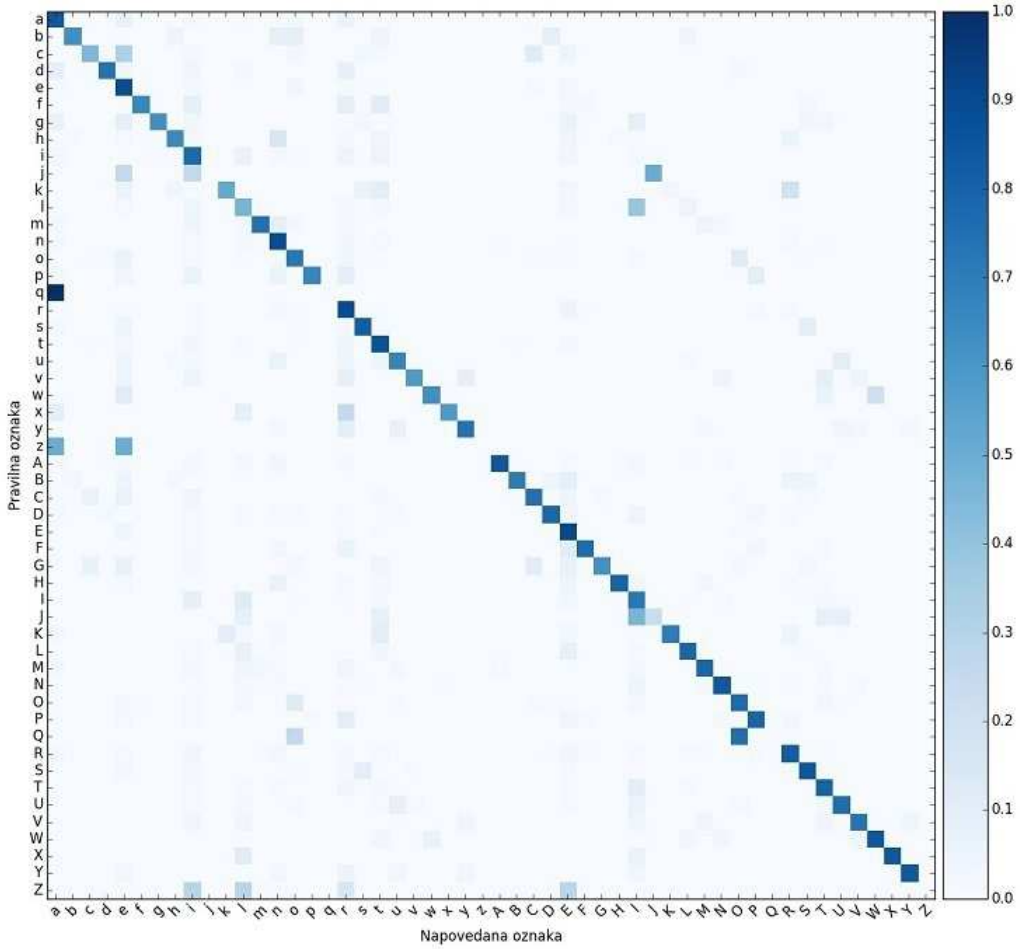
Sintetična zbirka slik znakov nam ni prinesla željenih rezultatov saj so se ti povprečno gibali v območju od 20 % do 25 % klasifikacijske točnosti. Pri klasifikaciji smo uporabili sintetično učno množico in jo evalvirali na testni učni množici zbirke ICDAR. Slabe klasifikacijske rezultate lahko pripišemo slabi reprodukciji naravnih scen, kjer smo z dokaj naivnimi metodami simuliranja naravnih efektov želeli imitirati slike naravnih scen. Na prvi pogled so primerki sintetične zbirke precej podobni primerom slik znakov ostalih podatkovnih zbirk, kot je razvidno na sliki 4.9, vendar je zaradi prevelike razlike med značilkami naravnih scen ter značilkami sintetičnih slik prišlo do prevelikih razlik, pri čemer smo prišli do ugotovitve, da tak tip sintetičnih slik znakov

klasifikacijskega modela ne nauči primerno.

S pomočjo matrike napak, ki ponazarja število napačno klasificiranih slik znakov, smo lahko identificirali specifične skupine znakov, ki so pogosteje razpoznane napačno. Matrika je rezultat vizualizacije uspešnosti algoritma HOG in klasifikatorja SVM (Chi-Squared) na podatkovni zbirki slik teksta ICDAR. Na podlagi slike 6.10, ki predstavlja klasifikacijo slik znakov, kjer je velikost znaka obravnavana kot nerelevantna, lahko razberemo, da je najpogosteje napačno klasificiran znak "q" z znakom "o", s približno 50 % napačnimi klasifikacijami, druga najpogostejša zamenjava znaka "q" je z znakom "a", v tem primeru je procent napačnih klasifikacij 40 %. Sklepamo lahko, da je zaradi invariantnosti velikosti znaka, mali znak "q" napačno klasificiran za mali "a". Poleg znaka "q" je drugi najpogosteje napačno klasificiran znak, znak "j", ki je zamenjan z znakom "i" v 35 % primerih. Iz matrike je razvidno tudi, da so najbolj natančno klasificirani znaki "a,", "e", "n", "o", "r", "s", "w", torej znaki katerih oblika je nedvoumna in bolj unikatna. Ti znaki so klasificirani pravilno v več kot 85 % primerih. Iz matrike napak pri variantnosti velikosti znaka 6.10, je še bolj očitno da gre za mali znak "q", ki je skoraj v vseh primerih zamenjan za znak "A", medtem ko je velik znak "Q" najpogosteje zamenjan za velik znak "O" in v nekaterih primerih tudi za mali znak "o". Razvidno je tudi, da je znak "z" najpogosteje napačno klasificiran kot "a" in "e" in znak mali "j" za velik "J", kar dejansko ni problematično, ker gre za enako črko. Med drugim je najpogosteje zamenjan znak "Q" z znakom "O", saj je med njima velika strukturna podobnost.



Slika 6.10: Matrika napak (angl. confusion matrix) pri invariantnosti velikosti znaka.



Slika 6.11: Matrika napak (angl. confusion matrix) pri razlikovanju velikosti znaka.

Poglavje 7

Zaključek

V magistrskem delu smo implementirali in analizirali algoritme za razpoznavo znakov v slikah naravnih scen in opravili evalvacijo na večih podatkovnih zbirkah slik teksta (ICDAR, Chars74K, CVL OCR DB) z različnimi klasifikatorji. Pri evalvaciji smo med seboj kombinirali algoritme za pridobivanje značilk (HOG, PHOG, Co-HOG) in različne klasifikacijske metode (SVM, K-NN, ANN, NB). Z analizo različnih parametrov na algoritmih za ekstrahiranje značilk 6.2 smo lahko dosegli optimalne klasifikacijske rezultate na klasifikacijskih algoritmih in podatkovnih zbirkah slik teksta. Prav tako smo z uporabo dobrih reprezentativnih vzorcev učne množice in večjim številom učnih vzorcev dosegali višjo klasifikacijsko točnost. Dokazali smo, da velikost učne množice korelira s klasifikacijsko točnostjo 6.9 ter pri tem identificirali optimalno velikost učne množice, ki naj bi štela od 9000 do 12000 učnih vzorcev.

Izkazalo se je, da najvišje klasifikacijske rezultate dosega algoritem PHOG v povezavi s klasifikacijsko metodo SVM in jedrom Chi-Squared, kjer je klasifikacijska točnost v povprečju za 1 % do 2 % višja od algoritma HOG z enakim klasifikatorjem. Višjo klasifikacijsko točnost tega algoritma lahko razložimo s tem, da algoritem PHOG generira vektorje značilk, ki so do 10-krat večji kot pri algoritmu HOG, kar pomeni, da hrani veliko več informacij o posamezni sliki znaka. Na podlagi tega smo ugotovili, da je algoritem HOG primernejši

za realnočasovno razpoznavo, saj je zaradi majhnega vektorja značilk procesorsko in časovno manj zahteven kot algoritem PHOG, ki temelji na piramidni shemi. Visoke klasifikacijske rezultate algoritma SVM, lahko pripišemo uporabi Chi-Squared jedra, ki zelo dobro obravnava vektorje značilk pridobljene iz histogramov, pri čemer so vrednosti vektorja pozitivne in normalizirane. Podpovprečne rezultate klasifikatorja NB lahko pripisujemo njegovi hitrosti, saj je zaradi računske nezahtevnosti hitrejši klasifikator v primerjavi z ostalimi (SVM, K-NN, ANN).

Področje optične razpoznave znakov se približuje natančnosti človeške razpoznave znakov. Povprečen človek naj bi na podatkovni zbirki ICDAR razpoznal znak z 92 % natančnostjo. Najsodobnejša metoda, CNN [10], ki je trenutno nov mejnik na področju razpoznave znakov in obravnava 62 klasifikacijskih razredov, dosega 84 % klasifikacijsko točnost na podatkovni zbirki slik teksta ICDAR. Tako visoka klasifikacijska natačnost je pogojena tudi z ogromno učno množico, ki šteje 50000 učnih vzorcev. Poleg ogromnega števila učnih vzorcev, so testni vzorci zbirke ICDAR, ki na slikah niso v celoti vidni, odstranjeni iz testne množice. V našem delu, je najvišjo klasifikacijsko natančnost dosegal algoritem PHOG v povezavi s klasifikacijskim algoritmom SVM in Chi-Squared jedrom in sicer 82,49 % klasifikacijsko točnost pri 26. razredih in 78,58 % točnost pri 52. razredih, na podatkovni zbirki slik ICDAR, kot je razvidno v tabeli 6.5. Pri tem je učna množica štela "le" 6000 vzorcev, kar je približno 8-krat manj kot pri evalvaciji metode CNN [10]. Omeniti moramo tudi, da se lahko določena količina napačno razpoznanih znakov prišteva nepravilni anotaciji znakov podatkovne zbirke slik teksta.

V nadaljnjem delu bi lahko, glede na graf slike 6.9, v kolikor bi povečevali velikost učne množice na 50000, dosegli še višjo klasifikacijsko točnost. Prav tako bi z uporabo naprednejših tehnik pridobivanja značilk kot so konvolucijske nevronske mreže in z izboljšanjem kvalitete učnih vzorcev slik teksta doprinesli k izboljšanju klasifikacijskih rezultatov.

Literatura

[1] (2016) Tesseract OCR. Dostopno na: https://github.com/tesseract ocr.

[2] (2016) ABBYY FineReader. Dostopno na: http://www.abbyy.com/finereader/about-ocr/what-is ocr/.

[3] (2016) The Street View Text Dataset. Dostopno na: http://vision.ucsd.edu/ kai/svt/.

[4] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, Y. Zu, W. Ou, C. Wolf, J. Jolion, L. Todoran, M. Worring, X. Lin. ICDAR 2003 Robust Reading Competitions: Entries, Results and Future Directions. *International Journal of Document Analysis and Recognition*, 7(2-3):str. 105–122, 2005.

[5] A. Ikica, P. Peer. Cvl ocr db, an annotated image database of text in natural scenes, and its usability. *Informacije MIDEM*, 41(2):str. 150–154, 2011.

[6] (2016) The Chars74K dataset. Dostopno na: http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/.

[7] Z. Daixian. Sift algorithm analysis and optimization. In *Image Analysis and Signal Processing (IASP), 2010 International Conference on*, pages 415–419, April 2010.

[8] M. Pomplun, W. Hsueh-Cheng. The attraction of visual attention to texts in real-world scenes. *Journal of Vision*, 12(6):26, 2012.

[9] Y. Feng, S. Uchida, R. Huang, P. Shivakumara. Scene character detection and recognition with cooperative multiple-hypothesis framework. *IEICE*, E96-D(10):2235–2244, October 2013.

[10] A. Coates, A. Y. Ng, T. Wang, D. J. Wu. End-to-end text recognition with convolutional neural networks. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 3304–3308, Nov 2012.

[11] F. Mamalet, P. Sebillot, K. Elagouni, C. Garcia. Combining multi-scale character recognition and linguistic knowledge for natural scene text ocr. In *Document Analysis Systems (DAS), 2012 10th IAPR International Workshop on*, pages 120–124, March 2012.

[12] D. James, G. Witten, R. Hastie, T. Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[13] P. Shivakumara, G. Louloudis, C. L. Tan, U. Pa,l S. Roy, P. P. Roy. Hmm-based multi oriented text recognition in natural scene image. In *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*, pages 288–292, Nov 2013.

[14] S. Susstrunk, G. Yildirim, R. Achanta. Text recognition in natural images using multiclass hough forests.

[15] J. Matas, L. Neumann. *Computer Vision – ACCV 2010: 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part III*, chapter A Method for Text Localization and Recognition in Real-World Images, pages 770–783. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[16] J. J. Yebes, S. Bronte, A. Gonzalez, L. M. Bergasa. A character recognition method in natural scene images. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 621–624, Nov 2012.

[17] B. Triggs, N. Dalal. Histograms of oriented gradients for human detection. In *In CVPR*, pages 886–893, 2005.

[18] D. Chen, J. M. Dobez. Robust video text segmentation and recognition with multiple hypotheses. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 2, pages II–433–Ii–436 vol.2, 2002.

[19] T. Gevers, K. Sezer, J. C. Gemert. *Computer Vision – ECCV 2012. Workshops and Demonstrations: Florence, Italy, October 7-13, 2012, Proceedings, Part III*, chapter Object Reading: Text Recognition for Object Recognition, pages 456–465. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[20] K. H. Ghazali, M. M. Mustafa, A. Hussain. Feature extraction technique using SIFT keypoint descriptors. *In Proceedings of the International Conference on Electrical Engineering and Informatics*, Electronic and Systems Engineering Faculty of Engineering, Universiti Kebangsaan Malaysia, Malaysia, 2007.

[21] (2016) Python. Dostopno na: https://www.python.org/.

[22] (2016) Cython. Dostopno na: http://cython.org/.

[23] (2016) PyPy. Dostopno na: http://pypy.org/.

[24] (2016) OpenCV. Dostopno na: http://opencv.org/.

[25] (2016) Numpy. Dostopno na:
http://www.numpy.org/.

[26] (2016) Matlab. Dostopno na:
http://www.mathworks.com/.

[27] (2016) SciPy. Dostopno na:
http://www.scipy.org/.

[28] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[29] (2016) Scikit learn. Dostopno na:
http://scikit learn.org/stable/.

[30] (2016) Matplotlib. Dostopno na:
http://matplotlib.org/.

[31] (2016) The Chars74K dataset. Dostopno na:
http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/.

[32] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, June 1986.

[33] O. R. Vincent, O. Folorunso. Folorunso, a descriptive algorithm for sobel image edge detection. In *in proceedings of Informing Science & IT Education Conference (InSITE*, 2009.

[34] X. Munoz, A. Bosch, A. Zisserman. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, pages 401–408, New York, NY, USA, 2007. ACM.

[35] C. Ponce, J. Lazebnik, S. Schmid. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.

[36] Z. R. Tan, S. Tian, and C. L. Tan. Using pyramid of histogram of oriented gradients on natural scene text recognition. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 2629–2633, Oct 2014.

[37] B. Su, C. L. Tan, S. Tian, S. Lu. Scene text recognition using co-occurrence of histogram of oriented gradients. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 912–916, Aug 2013.

[38] K. Yokoi, T. Watanabe, S. Ito. Co-occurrence histograms of oriented gradients for human detection. *IPSJ Transactions on Computer Vision and Applications*, 2:39–47, 2010.

[39] V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995.